

# Diseño y fabricación de un controlador de videojuegos con interfaz USB y forma de arma



---

## Universidad de Valladolid

Escuela Técnica Superior de Ingenieros de Telecomunicación

Grado en Ingeniería de Tecnologías de Telecomunicación

*Desarrollado por:*

**Oscar González González**

*Bajo la tutela de:*

**Jesús M. Hernández Mangas**

*2 de Septiembre de 2021*

## **RESUMEN.**

A lo largo de este documento se mostrarán las diferentes fases relativas a la planificación, elaboración y desarrollo de un producto final, específicamente de un controlador diseñado para su utilización en videojuegos. Para ello se tendrán en cuenta diversos factores como objetivos, procesos técnicos de desarrollo a nivel de *software* y *hardware*, diseño y costes, tanto a nivel económico como en términos de personal humano.

Con la finalidad de dilucidar los pasos llevados a cabo durante el desarrollo del controlador basado en un arma, concretamente una pistola, se expondrán las distintas decisiones tomadas durante su elaboración en base a diferentes criterios, y se especificará la elección de diversos parámetros a nivel técnico.

## **ABSTRACT.**

Throughout this document the different phases related to the planning, elaboration and development of a final product, specifically a controller designed to be used in video games, will be described. For this purpose, several factors will be taken into account, such as objectives, technical development processes at software and hardware level, design and costs, both in economic terms and in terms of human resources.

In order to elucidate the steps carried out during the development of the controller based on a weapon, specifically a pistol, the different decisions taken during its development based on different criteria will be presented, and the choice of various parameters at the technical level will be specified.

## **PALABRAS CLAVE.**

Producto, microcontrolador, circuito, figura, hoja de datos, sistema, desarrollo.

## **KEY WORDS.**

Product, microcontroller, circuit, figure, datasheet, system, development.

# ÍNDICE

<b>1. INTRODUCCIÓN Y OBJETIVOS.....</b>	<b>5</b>
<b>2. ESTUDIO GENERAL DEL SISTEMA.....</b>	<b>7</b>
2.1 Operating System (OS).....	7
2.2 HID.....	8
2.3 USB.....	8
2.4 Microcontrolador.....	9
2.5 Actuadores.....	9
2.6 Alimentación.....	9
2.7 Entrada Usuario.....	10
2.8 Sensores.....	10
<b>3. HARDWARE.....</b>	<b>12</b>
3.1 Búsqueda de componentes.....	12
3.1.1 Sensaciones del usuario.....	12
3.1.1.2 Interruptores de final de carrera:.....	15
Estudio teórico.....	15
Implementación práctica.....	16
3.1.1.3 LED.....	17
Estudio teórico.....	17
Implementación práctica.....	18
3.1.1.4 Electroimán.....	20
Estudio teórico.....	20
Implementación práctica.....	21
3.1.1.5 Motor descompensado.....	22
Estudio teórico.....	22
Implementación práctica.....	22
3.1.2 Sensores.....	24
Estudio teórico.....	24
Implementación práctica.....	26
3.1.3 Microcontrolador.....	27
Estudio teórico.....	27
Implementación práctica.....	31
3.2 Desarrollo electrónico esquemático.....	31
3.2.1 Análisis general de alimentación y conexiones.....	32
3.2.1.1 Alimentación.....	32
3.2.1.3 Conexiones.....	35
LED.....	36
Interruptores.....	36

Display LED.....	36
Botones.....	37
Motores .....	37
Sensores.....	39
3.2.2 Implementación del desarrollo electrónico esquemático .....	40
3.2.2.1 Configuración de las conexiones principales del microcontrolador .....	42
Entradas Digitales.....	42
Funciones alternativas de los puertos .....	43
Conexiones USB .....	44
3.2.2.2 Conexiones con sensores.....	46
3.2.2.3 Conexiones con motores .....	51
Resistencia.....	51
Diodo.....	52
Transistor.....	54
Regulador de tensión .....	57
3.2.2.4 Conexión con diodos LED .....	59
3.2.2.5 Conexión con <i>display</i> .....	60
3.2.2.6 Conexión con interruptores y botones.....	61
3.2.2.7 Otras conexiones y conectores .....	61
Bootloader.....	61
ISP.....	61
Conectores.....	62
3.2.2.8 Implementación final.....	62
3.3 Diseño de la PCB.....	64
3.4 Análisis de potencia.....	66
3.4.1 Conjunto de sensores MEMS.....	66
3.4.2 Motor vibrador.....	67
3.4.2.1 Regulador de tensión.....	67
3.4.2.2 Transistor 2N7002.....	68
3.4.3 Motor solenoide .....	69
3.4.2.2 Transistor NDS331N.....	69
3.4.4 Microcontrolador .....	70
3.4.4.1 Salidas de datos .....	71
3.4.4.2 Entradas de datos de interruptores y botones .....	71
3.4.4.3 Salidas hacia diodos .....	71
3.4.4.4 Salidas PWM para los motores .....	72
3.4.4.5 Otras consideraciones.....	72
3.4.5 Cómputo global y conclusiones .....	72
3.5 Obtención de materiales .....	74
3.5.1 BOM .....	74

3.5.2 Adquisición de PCB.....	77
3.6 Montaje de componentes .....	79
<b>4. SOFTWARE.....</b>	<b>81</b>
4.1 Instalación del gestor de arranque .....	81
4.2 Bibliotecas empleadas .....	83
4.2.1 AbsMouse.h .....	83
4.2.2 Keyboard.h.....	86
4.2.3 MPU9250.h.....	86
4.2.4 SevenSeg.h.....	88
4.2.5 SoftwareReset.h .....	89
4.3 Desarrollo teórico del código implementado .....	89
4.3.1 Estructura general del código fuente.....	89
4.3.1.1 Setup.....	90
4.3.1.2 Loop .....	90
4.3.2 Desarrollo del algoritmo de posicionamiento .....	91
4.3.2.1 Fundamentos matemáticos .....	92
<b>5. DISEÑO DE LA CARCASA.....</b>	<b>96</b>
5.1 Desarrollo del diseño tridimensional .....	97
5.2 Configuraciones de impresión .....	104
5.3 Postprocesado .....	107
<b>6. COSTES HUMANOS .....</b>	<b>109</b>
<b>7. IMPLEMENTACIÓN FINAL.....</b>	<b>112</b>
<b>SIGLAS. ....</b>	<b>114</b>
<b>REFERENCIAS. ....</b>	<b>116</b>
<b>ANEXO .....</b>	<b>121</b>
1.Obtención de materiales .....	121
2. Código implementado.....	128
3. Manual de usuario .....	141

# 1. INTRODUCCIÓN Y OBJETIVOS

Este proyecto tiene como fin llevar a cabo el estudio, desarrollo e implementación de un dispositivo apuntador, con el propósito de su empleo en entornos recreativos. Para desempeñar esta labor se tratarán de establecer unos objetivos que servirán como bases del proyecto planteado, además de describir algunas de las nociones generales necesarias para poder realizar los diferentes apartados que conforman el grueso del proyecto.

Se citan a continuación los objetivos:

- Analizar las distintas fases en la realización de un producto electrónico, así como la forma de ponerlas en práctica.
- Adquirir conocimientos en cuanto a procesos de desarrollo electrónico.
- Ser capaz de utilizar herramientas *Computer Assisted Design* (CAD) con el fin de realizar esquemas de circuitos, consiguiendo agilizar el desarrollo del producto.
- Exponer con claridad los criterios empleados a la hora de tomar decisiones sobre los elementos que formarán parte del proyecto.
- Conocer y mitigar los posibles efectos negativos que puedan surgir durante la realización electrónica, tales como efectos térmicos de los componentes, distribución de los mismos, definición del cableado, coste energético, etc.
- Conseguir información acerca de la obtención de un circuito impreso con el fin de obtener su implementación física.
- Gestión *software* del *hardware* creado desde su detección hasta su programación de más alto nivel.
- Alcanzar el propósito final de realización de un producto desde cero, documentando el proceso llevado a cabo y los pasos seguidos durante su desarrollo.

Si tratamos de realizar una descripción del proyecto más exhaustiva tendremos que complementar los anteriores objetivos con una definición que goce de un mayor grado de especificidad.

En primer lugar, el dispositivo apuntador será concebido como un control externo que permitirá su utilización en modos de juego de primera persona, orientados a disparos o *shooters*. Para ello el producto tendrá que contar con una serie de elementos que permitan al usuario reconocer que está utilizando algo similar a una herramienta de disparo, en

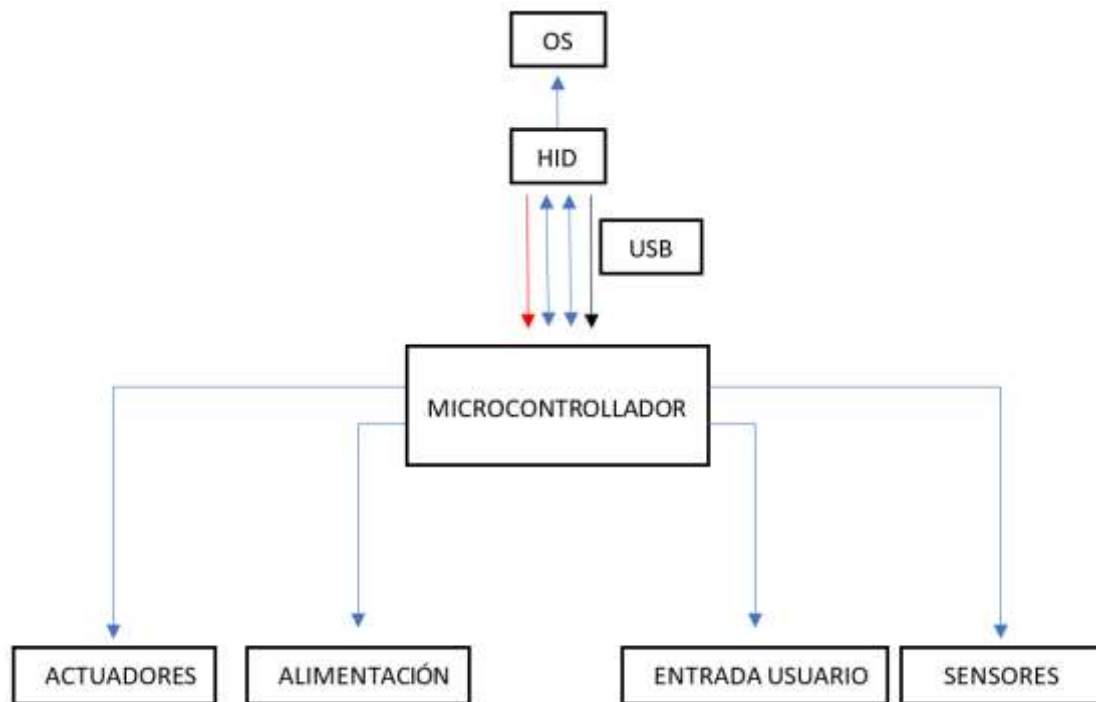
nuestro caso utilizaremos un concepto básico de partida centrándonos en el desarrollo de un dispositivo que cuente con las características de una pistola.

Para conseguir adaptar la implementación del producto a tal fin, el dispositivo contará con diversos elementos que faciliten su reconocimiento, como una cubierta plástica que se asemeje a tal objeto de la vida real. Pero también con otros elementos que puedan llegar a producir unas respuestas sensoriales semejantes. En el caso de este último objetivo, se emplearán útiles electrónicos variados, como actuadores y sensores.

A mayores de estos componentes electrónicos, será necesario un microcontrolador que los gestione y realice un control sobre los momentos de activación de estos, además de encargarse de la definición de sus comportamientos ante diversas situaciones.

## 2. ESTUDIO GENERAL DEL SISTEMA

Una vez citados estos objetivos, hemos de centrarnos en cómo será la puesta en marcha del proyecto. Inicialmente tendrá que plantearse de manera esquemática y conceptual cuáles serán los módulos necesarios para alcanzar los objetivos. En la **Figura 1** se muestra un esquema representativo sobre los fundamentos de la electrónica y la interfaz de comunicaciones del proyecto.



*Figura 1. Esquema conceptual del controlador.*

En cuanto al estudio del sistema, pasaremos a establecer una serie de especificidades sobre los distintos elementos modulares mostrados en el esquema de la **Figura 1**. Así como explicar el porqué de los distintos módulos empleados y los criterios que nos han llevado a elegirlos.

### 2.1 Operating System (OS)

La finalidad que se persigue con este proyecto es que nuestro dispositivo sea capaz de utilizarse en todo tipo de sistemas operativos o en su defecto en el mayor número posible. Debido a esta premisa, nuestro dispositivo tendrá que ser compatible con *Human Interface Devices* (HID).



## 2.2 HID

La idea detrás de estas siglas no es más que el hecho de comprender la necesidad de un dispositivo que traduzca la información que los seres humanos aportamos a los ordenadores y otros aparatos electrónicos. El concepto es muy básico, pero tiene como fin una estandarización, mediante la cual diferentes fabricantes de dispositivos puedan compatibilizar sus productos con una amplia gama de sistemas.

Cuando nos referimos al concepto de HID, en realidad estamos hablando de la clase *Universal Serial Bus* (USB) HID y es que la clase USB HID surge con el objetivo de proporcionar a los fabricantes de HID la información necesaria para crear dispositivos compatibles con USB. Además, se fundamenta en las premisas de: ser lo más compacto posible para ahorrar espacio de datos en el dispositivo, permitir que la aplicación de software omita información desconocida, ser extensible y robusto y ser autodescriptivo para permitir aplicaciones de software genéricas [1]. Se puede obtener más información sobre la definición de HID en [2].

Con la certeza de utilización de HID, debido a su uso masivo y a las facilidades que aporta, nos vemos directamente expuestos a USB.

## 2.3 USB

Es un estándar que define las comunicaciones en un *Bus* de datos, así como los cables empleados, los conectores y los protocolos. Fue creado por un grupo de empresas del sector tecnológico que vieron la necesidad de unificar el proceso de conexión de periféricos [3]. Además, como su propio nombre indica, transmite datos en serie. Actualmente se ha convertido, casi de forma completa, en el estándar de facto en lo referente a las conexiones de periféricos.

USB posee un cableado con cuatro pines diferentes, de los cuales dos son de datos y otros dos de alimentación, esto último puede verse reflejado en el diagrama de la *Figura 1*.

Es importante hacer hincapié en sus cuatro cables de conexión, ya que será necesario hacer una toma de decisión en cuanto a la comunicación con USB y determinar si la alimentación del sistema vendrá únicamente proporcionada por el conector USB o si, por el contrario, será necesario dotar al conjunto de una alimentación independiente a mayores.

## 2.4 Microcontrolador

Una vez establecidos los anteriores parámetros, será necesario aportar a nuestro proyecto su principal componente, el microcontrolador. Este microcontrolador será el interlocutor primario entre usuario y ordenador. Este, ha de ser compatible con los recursos necesarios que se han determinado tras la definición de objetivos en un inicio: tendrá que ser compatible con el protocolo de comunicaciones USB y además tendrá que poder identificarse dentro del sistema como un dispositivo señalador HID.

A mayores de cumplir los anteriores requisitos, el microcontrolador deberá contar con una frecuencia de trabajo que nos permita crear un dispositivo señalador fluido, así como la capacidad de gestionar el resto de elementos del circuito, desde los actuadores hasta los sensores.

El microcontrolador será uno de los principales pilares en el desarrollo del producto, ya que de él dependerán, en gran parte, las limitaciones del controlador en su conjunto, tales como consumo de potencia, máximas tasas de muestreo, alimentación, velocidad de procesamiento...

## 2.5 Actuadores

Nos permitirán transformar la energía eléctrica en magnitudes físicas que proporcionarán diferentes estímulos en el usuario. Estos actuadores serán elementos que produzcan movimiento físico, de forma que doten de realismo a nuestro dispositivo aportándole experiencias sensoriales que aumenten el nivel de inmersión durante el empleo del mismo. Algunos de estos estímulos podrían estar relacionados con la vibración, o retroceso. Los actuadores serán gestionados desde el microcontrolador, con el propósito de activarlos en momentos definidos en los que se realicen determinadas acciones como apretar el gatillo.

## 2.6 Alimentación

La alimentación será la encargada de aportar potencia a nuestro sistema. De manera que deberemos determinar cuál es la potencia que utilizarán los distintos elementos: si

necesitaremos etapas de adaptación de alimentación (transformadores, amplificadores) o si, por el contrario, tendremos que utilizar varias fuentes independientes.

Además, esta alimentación tendrá que compatibilizarse con la alimentación del conector USB que tendrá distintos valores en función de la versión de este protocolo. Las decisiones entorno a la alimentación vendrán delimitadas por los elementos empleados.

## **2.7 Entrada Usuario**

Únicamente se han descrito, hasta ahora, elementos propios del sistema o que provocarán sensaciones en el usuario. A pesar de esto se hace necesaria la realimentación de información al sistema por parte del usuario. Por este motivo, será preciso incorporar mecanismos como botones que nos faciliten la tarea a la hora de conocer las intenciones del usuario y permitan realizar distintas configuraciones externas en nuestro dispositivo.

## **2.8 Sensores**

Los sensores jugarán un papel fundamental en el dispositivo apuntador y es que serán los que doten de información al microcontrolador para que este, una vez procesados los datos, pueda dar indicaciones al equipo en el cual esté conectado.

Su función principal será la recopilación de información. Más específicamente, tendrán que proporcionarnos datos que nos permitan definir una posición en una pantalla en dos dimensiones. De manera que, una de las opciones podría ser un acelerómetro y un giroscopio, por relacionarse de manera relativamente sencilla las magnitudes físicas medidas en estos, con el espacio.

Nosotros necesitaremos relacionar el espacio en dos dimensiones con las magnitudes físicas que captan estos sensores, de manera que, a través de la programación seamos capaces de convertir esos datos en los datos que nosotros queremos representar por pantalla.

También se podrían incluir otros sensores para realizar una calibración del dispositivo o simplemente con el fin de posibles actualizaciones mejorando así la escalabilidad del producto.

Como comentario final, queda argumentar que los elementos externos al equipo, es decir, el microcontrolador, los sensores y los actuadores, tendrán que poseer un diseño compacto que permita su ensamblado en la carcasa plástica, de manera que se buscará crear un diseño adaptado a las necesidades globales.

Con esto se podrán cumplir los objetivos relacionados con el aprendizaje sobre los procesos de un desarrollo electrónico y la utilización de herramientas CAD para la realización final del circuito.

## **3. HARDWARE**

### **3.1 Búsqueda de componentes**

Tras describir a grandes rasgos el estudio sobre el sistema, debemos realizar una discusión sobre los distintos componentes electrónicos que formarán parte de nuestro dispositivo.

En el estudio del sistema se definieron de manera genérica algunos de los elementos. Estos elementos explicaron de forma conceptual las necesidades que debería solventar la implementación de los componentes electrónicos, es por esto que, se llevará a cabo una discusión sobre los componentes.

Dado que el microcontrolador será el centro neurálgico de nuestro dispositivo, tendrá que adaptarse al resto de elementos que empleemos durante nuestra elaboración, así como a las características definidas en el estudio del sistema, motivo por el cual se definirá en último lugar.

#### **3.1.1 Sensaciones del usuario**

Los actuadores serán el pilar de los estímulos externos captados por el usuario, de manera que tendremos que elegirlos en base a las prestaciones que queremos ofrecer en nuestro dispositivo.

Con el fin de realizar una implementación lo más realista posible, se pasará a enumerar cuáles serán los parámetros que definan la experiencia de usuario en una situación real, para después intentar emular los mismos en nuestro producto, mediante distintos actuadores. En la vida real el disparo de una pistola produce diferentes efectos.

Cuando esta no se utiliza no produce ningún estímulo al portador, el único instante en el que ocurren los acontecimientos más notorios que provocan realimentación en el usuario es tras el disparo de esta. Al emplear una pistola:

1. Inicialmente la persona que la utiliza tiene que colocar las balas en el cargador e introducirlo en su interior, después de esto deberá deslizar la corredera de la pistola para que la bala se desplace del cargador a la recámara.

2. Una vez la bala está en la recámara, el usuario se encontrará con una resistencia, en forma de fuerza, al apretar el gatillo, debido a que para efectuar el disparo tendrá que accionar el martillo.
3. Tras actuar el gatillo, el usuario percibirá un destello producido por la salida de los diferentes gases expulsados a través del cañón. Esto es consecuencia de la reacción química originada por la fuerza ejercida por medio del percutor del martillo en el fulminante de la bala y la pólvora de su interior.
4. Acto seguido a la explosión será notable el retroceso, este retroceso tiene su origen en un proceso físico que se puede explicar mediante la tercera ley de Newton. La cual dictamina que, si un objeto ejerce una fuerza sobre otro, entonces el segundo objeto debe ejercer una fuerza de igual magnitud en dirección opuesta sobre el objeto inicial [4]. Como consecuencia de este retroceso la corredera es impulsada hacia atrás de manera que el casquillo de la bala pueda ser liberado. Al contar con un muelle en el mecanismo de la corredera, esta volverá a su posición inicial e introducirá una nueva bala en la recámara, permitiendo así replicar el proceso de manera consecutiva hasta vaciar el cargador y pudiendo volver nuevamente al paso inicial.

Una vez explicado brevemente este proceso, nos centraremos en la forma de emular estas sensaciones en nuestro producto de manera análoga. Con lo cual, en nuestro producto:

1. En primer lugar, al estar destinada a un entorno recreativo, nuestra pistola no tendrá ningún sistema mecánico que imite el cargador o las balas, pues sería tedioso contar con estos elementos y tener que llenar el cargador cada vez que quisiésemos obtener más munición. Además, el sistema de munición ya viene implementado en muchos videojuegos de manera automática, provocando una utilización más amigable. Sin embargo, sí será posible implementar el movimiento de la corredera inicial, el cuál será representado con un interruptor de final de carrera que permita verificar el hecho de que la corredera ha llegado al final de su trayectoria, consiguiendo introducir la primera bala en la recámara.
2. Continuando con la acción del gatillo, se introducirá nuevamente un interruptor de final de carrera que permita comprobar que se ha accionado el gatillo lo suficiente como para que se libere el martillo.

3. En lo referente al destello producido por la salida de los gases del cañón, se tratarán de introducir diversos *Light-Emitting Diode* (LED) que, con una configuración grupal de los mismos, aporten una luminancia elevada.
4. Finalmente, con el propósito de asemejar tanto el retroceso de la corredera tras la explosión, como el retroceso físico del arma, se introducirán dos actuadores. El retroceso de la corredera será llevado a cabo por un electroimán, concretamente un motor solenoide que cuente con un muelle para así poder devolver la corredera a su lugar original. En cuanto al retroceso físico del arma es mucho más complejo emularlo, dado que no podremos moverla de las manos del usuario, pero sí que podremos introducir un motor descompensado que actúe como vibrador, con el fin de reproducir la inestabilidad del objeto en las manos de este.

Para finalizar con la argumentación relativa a la emulación en nuestro dispositivo, cabe explicar que nuestro producto estará centrado, principalmente, en pistolas automáticas o semiautomáticas, es decir, no será necesario desplazar la corredera para introducir la bala del cargador en la recámara tras cada disparo, sino que bastará con introducir la primera bala de un nuevo cargador, salvo que se desee operar en modo manual. De esta manera podremos asociar la acción de recargar el arma a la incorporación de la primera bala del cargador en la recámara.

En cuanto a las funciones de modo automático o semiautomático vendrán implementadas por *software* y serán gestionadas por el microcontrolador, aunque será necesario introducir un pequeño conmutador que permita cambiar el modo del arma, o cierta configuración *software* que permita seleccionar el modo automático.

Será importante tener en cuenta que el retroceso en la corredera ejercido por el motor solenoide no deberá activar el interruptor de final de carrera de esta, al contrario de lo que ocurre en la vida real. Esto sucede porque de esta manera podremos asociar la acción de introducir un nuevo cargador a la recarga en los juegos, utilizaremos el interruptor de final de carrera como botón para este fin.

Una vez expuestas las necesidades y citados los elementos que nos permitirán llevar a buen puerto el desarrollo de nuestro producto, pasaremos a explicar de manera detallada los componentes específicos que llevarán a cabo estas funciones.

Cabe destacar que, durante el estudio teórico de cada componente, se ha realizado la búsqueda de elementos que contasen con características tangibles capaces de justificar sus implementaciones a nivel técnico. Como consecuencia, los elementos teóricos elegidos durante la búsqueda de componentes cuentan con hojas de datos o *datasheet* rigurosos y detallados. Por el contrario, a la hora de intentar acceder a estos elementos específicos en el mercado global, ha sido prácticamente imposible encontrar los modelos concretos. Debido a este motivo, se ha recurrido a componentes similares, que en muchas ocasiones no contaban con información detallada sobre sus características electrónicas, pero que ofrecían facilidad de adquisición y disminuían los costes considerablemente, además de ajustarse a los tiempos fijados durante el desarrollo del producto.

A continuación, se aportará una discusión detallada sobre el proceso de selección de los elementos principales del circuito, en relación con el estudio general del sistema. Aquellos componentes que poseen una utilización más generalizada como botones y conmutadores quedarán fuera del siguiente punto, así como otros elementos electrónicos que serán presentados durante las siguientes secciones.

En esta discusión se aportarán tanto los argumentos teóricos planteados como la implementación final de los mismos. Esta variación se debe a distintos aspectos, desde motivos económicos o tiempos de envío elevados, hasta cese de fabricación o falta de *stock* de diversos componentes. Como consecuencia, en algunos de los casos no ha sido posible encontrar hojas técnicas de datos que relaten las características específicas de los elementos implementados y se han elegido en función de las descripciones aportadas por los distintos vendedores que proporcionan los productos.

### **3.1.1.2 Interruptores de final de carrera:**

#### *Estudio teórico*

Los interruptores de final de carrera que utilicemos deberán ser de un tamaño relativamente pequeño, con el fin de no sobredimensionar la carcasa del producto. Para nuestra aplicación se ha determinado que necesitará contar con un cabezal de rodillo en el caso del gatillo, dado que este elemento se accionará un número muy elevado de veces y podría provocar fatiga en el dedo en caso de necesitar ejercer demasiada presión para accionarlo. Por el contrario, la corredera se empleará con menor frecuencia y podrá contar con un interruptor algo más rígido, de esta manera se aportará mayor realismo a la



corredera. Ambos interruptores deberán tener una durabilidad mecánicamente elevada, como consecuencia al reiterado accionamiento de los mismos.

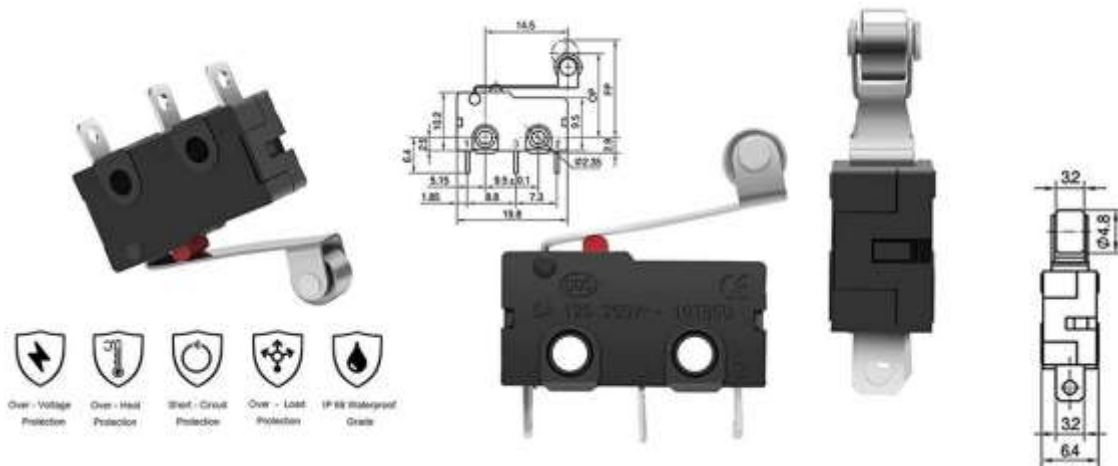
Es por esto que se han elegido los interruptores de final de carrera **DG13-B1RA**[5]. En el caso del gatillo contará con un cabezal auxiliar de rodillo que provocará que sea necesaria menos fuerza para activarlo y de esta manera se obtenga un accionamiento temprano del interruptor. En el caso de la corredera no se incorporará ningún cabezal obligando a moverla de manera completa para accionar el interruptor.

Sus dimensiones son pequeñas “*Very small size (only 12.8 x 5.8 x 6.5 mm)*” [5] y podrán encajar fácilmente en nuestro diseño, además soportan altos valores de tensión y corriente “*the breaking capacity ranges from small switching currents and voltages to low-voltage applications up to 3 A 125 V AC*”[5] previniéndonos de quemar el interruptor en caso de una mala conexión de diseño. También poseen tres patillas *Single Pole Double Throw* (SPDT) aunque solo utilicemos dos de ellas, la tercera patilla nos podría servir para implementaciones futuras, mejorando la escalabilidad del producto. Este componente cuenta además con una vida mecánica alta “*depending on the model > 1.000.000 actuations*”[5].

#### ***Implementación práctica***

A la hora de adquirir este elemento se han encontrado problemas de disponibilidad y se ha obtenido una alternativa viable de cara a su implementación final. El modelo elegido ha sido el modelo **T403** de la empresa GTIWUNG. La principal característica en la que dista el nuevo componente del interruptor de carrera anteriormente documentado, es su tamaño, poseyendo unas medidas de 21 x 24.1 x 6.4mm [6] frente a los 12.8 x 5.8 x 6.5mm (sin tener en cuenta patillas ni palancas). En relación con el planteamiento teórico anterior, podemos establecer múltiples similitudes. En este caso, nos encontramos nuevamente frente a un interruptor SPDT [6], el cual cuenta con una mayor tolerancia frente a altos valores de tensión y corriente, siendo estos de 5A y 125V-250V [6]. Respecto a sus dimensiones, podemos observar que estas superan el tamaño del componente estudiado, sin embargo, estos incrementos no serán muy significativos y podrán adaptarse en el desarrollo de la carcasa del producto sin producir graves problemas de diseño. De cara a su durabilidad, no encontramos valores proporcionados en la página del vendedor, aunque en contraposición se aportan algunas características del

componente como protección contra sobrecargas y calentamiento, según se observa en la *Figura 2*.



*Figura 2.* Características y dimensiones aportadas por el vendedor del componente T403 de la empresa GTIWUNG. [6]

### 3.1.1.3 LED

#### *Estudio teórico*

Para la selección de este componente deberemos tener en cuenta que estamos buscando un diodo emisor de luz que goce de una luminancia elevada, tratando así de imitar de una manera más fidedigna los destellos provocados por la pólvora. A mayores, el color emitido por el LED tendrá que ser similar al de una explosión, por esto se ha elegido un color ámbar.

Como consecuencia, tras examinar su hoja de datos [7], se ha determinado que un buen candidato para desempeñar este papel sería el LED “**Super Bright Amber Yellow LED RF-WW05C3AYP4-N2**”. En la hoja de datos se aprecia que este diodo puede ser un buen candidato, debido a que soporta un máximo de 5V en polarización inversa y una corriente de hasta 100mA a temperatura ambiente. También se muestra cómo los destellos del LED podrán ser de alta frecuencia ya que en [7] se cita “*Peak Forward Current ( $\leq 1/10$  Duty Cycle, 0.1ms Pulse Wide)*” lo que significa que podríamos activar el diodo con un tren de pulsos de una frecuencia de hasta 10000Hz a la corriente máxima si en el ciclo de

trabajo no supera el 10% del ancho de pulso total, de esta manera no sobrecalentaremos el diodo, en nuestro caso estos son valores que cumplirán con creces nuestras necesidades.

En lo referente a la intensidad lumínica, es de 13.8 cd para una fotocorriente de 20mA, un valor algo bajo, sin embargo, se podrá utilizar en combinación con más LED idénticos, aumentando así estos valores. Bien es cierto que estos son los valores típicos y podrían conseguirse magnitudes mayores. Si realizamos los cálculos en lúmenes para hacernos una mejor idea del valor de intensidad luminosa a nivel comercial, observamos que, los lúmenes percibidos por el observador serían:

$$\Phi_v(\text{lm}) = I_v(\text{cd}) * \Omega(\text{sr}) = 13.8 * (2\pi(1 - \cos(25^\circ/2))) = 2.055324 \text{ lm}^1$$

El hecho de que estos valores no sean altos reside principalmente en su pequeño tamaño, también hay que tener en cuenta que buscamos simular este efecto, pero una intensidad lumínica demasiado elevada provocaría reflejos en la pantalla en la que se utilizase el dispositivo apuntador, empeorando la visión del puntero en esta.

En cuanto al color, este se asemeja bastante al de un disparo real, si consultamos las gráficas de la intensidad de emisión relativa en función de la longitud de onda, nos damos cuenta de que emite con un valor más elevado los colores correspondientes con las longitudes de onda entre 520-650nm correspondiendo con un amarillo un poco anaranjado.

Finalmente hablaremos de la directividad. Este diodo es bastante directivo “*Viewing Angle / Deg 25°*” [7], por lo que para crear la sensación de la explosión necesitaremos colocar varios de estos elementos formando un semicírculo con el fin ensanchar el ancho de haz global.

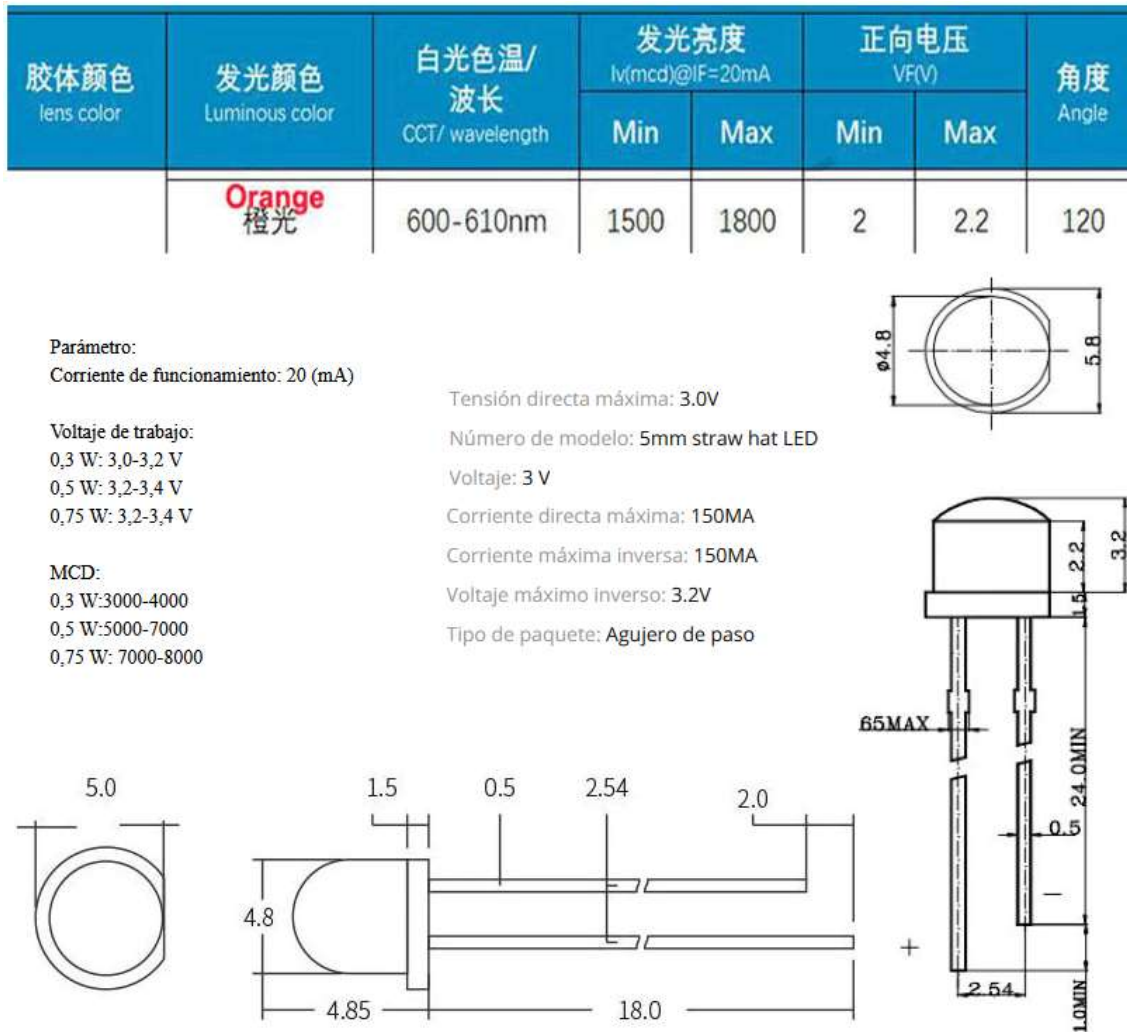
### ***Implementación práctica***

De cara a la obtención del modelo LED expuesto anteriormente, su adquisición no se ha realizado por motivos económicos. Este tipo de elementos electrónicos se corresponde de manera directa con un consumo masivo en el que los compradores no suelen presentar grandes exigencias específicas. A la hora de adquirir este componente se podría haber optado por emplear el componente expuesto durante el desarrollo teórico, sin embargo, el precio provocaría que nuestro producto tuviera un incremento en los costes de

---

<sup>1</sup> Dado que no se cuenta con una distancia fija de referencia respecto al observador, se ha supuesto para este cálculo que el área es igual al radio al cuadrado  $\frac{\text{altura}}{\text{radio}} = \frac{1}{2\pi} \xrightarrow{\text{Por tanto}} \Omega = 2\pi(1 - \cos\theta)\text{sr}$

producción como consecuencia de forzar unas características no demasiado notables en la experiencia final del usuario. Es por esto que se ha optado por utilizar un LED genérico que poseyera un brillo elevado, en este caso del vendedor KY WIN ROBOT [8]. En la **Figura 3** puede observarse un resumen con las principales características aportadas por el vendedor del producto.



**Figura 3.** Principales características aportadas por el vendedor del componente 5mm Straw Hat LED.[8]

La utilización de este componente permite ver algunas carencias frente al elemento teórico. La principal carencia será la intensidad luminosa, es decir, la potencia luminosa emitida por el LED en una dirección dada. En este caso, para una corriente de 20mA podremos obtener un máximo de 1800mcd [8], es decir 1.8cd, lo cual disminuye en un orden de magnitud las características del componente estudiado con anterioridad, sin embargo, como el ángulo es superior (el LED posee una directividad menor), el flujo

luminoso será mayor. Suponiendo que el ángulo proporcionado por el vendedor se expresa en grados sexagesimales:

$$\Phi_v(\text{lm}) = I_v(\text{cd}) * \Omega(\text{sr}) = 1.8 * ( 2\pi(1 - \cos(120^\circ/2)) ) = 5.654866 \text{ lm}$$

De esta forma estaremos consiguiendo un resultado óptimo para nuestro producto y es que podremos, a costa de sacrificar la intensidad luminosa en un punto específico, abarcar un ángulo de 360 grados en lo relativo a la iluminación. Dicho ángulo simulará el destello producido por la pólvora. Dado que no es muy directivo, bastarán 3 LED con un ancho de haz de 120° para cubrir la totalidad del destello circular.

Además, estaremos disminuyendo el consumo, dado que este diodo emisor de luz trabaja a la misma intensidad que el mostrado anteriormente, pero únicamente cuenta con un voltaje de polarización en inversa de 3V[8].

Como argumentos finales a favor, citar que el tamaño de la cúpula del LED en este caso es incluso inferior [8] a la del LED color ámbar seleccionado durante la búsqueda teórica. Además, este nuevo LED puede solicitarse en color naranja[8], aportando mayor realismo al efecto del destello de la explosión.

#### **3.1.1.4 Electroimán**

##### *Estudio teórico*

En el caso del motor solenoide, nos centraremos en su durabilidad y en la fuerza que genera, dado que tendrá que ser capaz de mover la corredera.

Este motor [9] funciona con un voltaje en continua de 5V soportando hasta 5.5W. En cuanto a su vida útil soporta hasta 300 000 ciclos, con un ciclo de trabajo de 1/4 para ciclos de 4 segundos, lo cual, quizás sea algo ínfimo ya que el producto accionará el motor solenoide cada vez que se active el gatillo y este último se presionará con mucha frecuencia. Además, habrá que tener en cuenta el modo de disparo automático que se quiere implementar, el cual será aún más agresivo. A pesar de esto, puede ser un punto de partida desde el que comenzar a modelar nuestro sistema.

En cuanto a la fuerza de atracción nos encontramos que es capaz de mover 80 gramos, valor aceptable ya que la impresión en tres dimensiones de piezas plásticas generalmente no superará esta magnitud, pero todo depende de las dimensiones del producto. Otro punto más negativo es que el recorrido del solenoide es de únicamente 3mm y se haría

insuficiente contar con un retroceso de este valor, sin embargo, sería posible añadir al montaje de la carcasa piezas plásticas mecánicas que solventaran este problema. Finalmente, sus dimensiones son adecuadas para implementarlas en un diseño compacto y funcional.

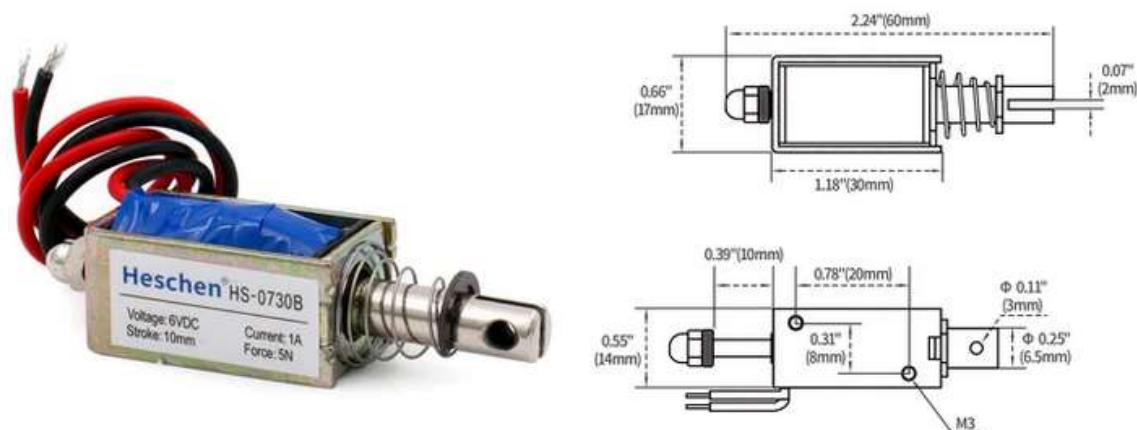
El motor elegido es el **ZH0-0420S-05A4.5** de la empresa SHENZHEN ZONHEN ELECTRIC APPLIANCES Co. , Ltd. En el caso de los motores solenoides, dado que se producen en masa y generalmente vienen producidos desde china, es una tarea difícil el encontrar fichas técnicas de especificaciones, por lo que este componente, a pesar de poseer algunas carencias, puede ser un buen punto de partida.

### *Implementación práctica*

A la hora de realizar la compra del motor solenoide empleado durante la búsqueda teórica no hemos sido capaces de encontrar un modelo que se adecuara a las condiciones requeridas de coste temporal y económico. Es por esto, que se ha seleccionado un nuevo modelo, el cual incorpora algunas características que mejorarán la experiencia de usuario.

Se trata del motor solenoide **HS-0730B** producido por la empresa Heschen [10]. La empresa no aporta ningún documento en el que figuren sus características, sin embargo, arroja luz sobre algunas propiedades del solenoide, estas pueden observarse en la

**Figura 4.**



**Figura 4.** Características del motor solenoide HS-0730B aportadas por el fabricante.[10]

Tal y como se muestra en la anterior figura, el motor solenoide posee unas características similares al utilizado durante la búsqueda de componentes, compartiendo con el solenoide **ZHO-0420S-05A4.5** su fuerza y corriente de operación.

En cuanto al voltaje, este solenoide trabajará a un voltaje de 6V, aunque podrá ser utilizado a 5V con la misma intensidad. Al no existir una hoja de datos que lo respalde podremos intuir que quizás disminuya la fuerza generada por el motor, pero no de manera significativa (Se ha verificado de manera experimental que el electroimán permite su actuación a un voltaje de 4.5V y 1A). En cuanto a las dimensiones, es cierto que es bastante superior al anterior motor solenoide, sin embargo, podría ajustarse sin problema en el diseño tridimensional. Como ventaja citar que, este nuevo componente posee un desplazamiento de 1mm, lo que provocará un retroceso mayor en la corredera, aportando un realismo superior al producto.

### 3.1.1.5 Motor descompensado

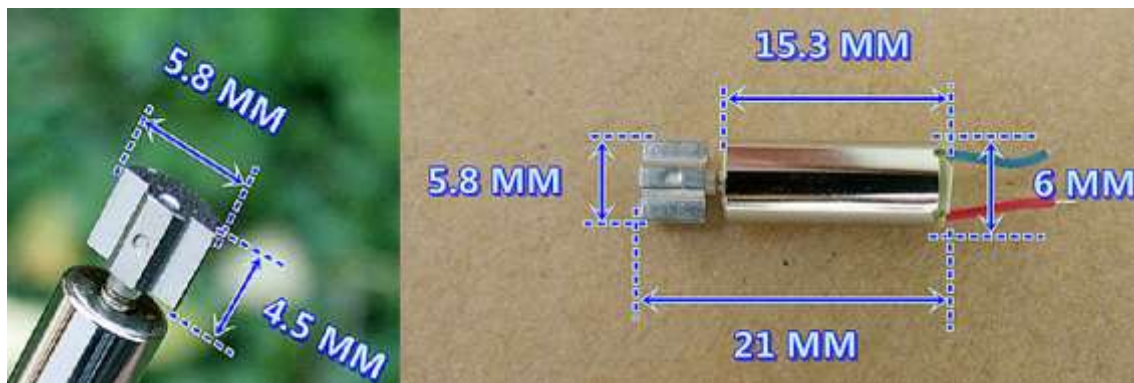
#### *Estudio teórico*

Para el motor vibrador se ha seleccionado el modelo **B1034.FL45-00-015** de la empresa ZHEJIANG YUESUI ELECTRON STOCK CO., LTD. Principalmente ha sido seleccionado por su ínfimo tamaño, y es que únicamente ocupa 1cm, además de esto posee una estructura circular o de moneda que facilitará su integración dentro de la placa.

En lo referente a sus características electrónicas funciona entre 2.3-3.6 V de tensión en continua y soporta una corriente máxima de 60mA. Estos valores son bastante pequeños y no repercutirán de manera muy significativa en el consumo. A pesar de esto, este motor consigue alcanzar las 13000rpm. Otra de las características notables de este motor vibrador es que no posee un calentamiento elevado, tal y como se refleja en su hoja de datos [11].

#### *Implementación práctica*

En el caso de este componente se ha tenido que recurrir a un motor vibrador genérico que se asemejará en dimensiones, aunque modificará su funcionamiento respecto al motor vibrador elegido durante la búsqueda de componentes. En este caso, nos encontramos con un motor sin núcleo, pensado principalmente para productos como cepillos de dientes. En la **Figura 5** se muestran las dimensiones del motor seleccionado.



*Figura 5. Dimensiones del motor vibrador aportadas por el vendedor.[12]*

Si comparamos las características de este producto con el motor vibrador **B1034.FL45-00-015**, podremos apreciar que cuenta con características similares.

En primer lugar, aunque el formato de forma “de moneda” o redondo cambia, el tamaño sigue siendo reducido. Tal y como figura en su hoja de datos [11], el motor **B1034.FL45-00-015** cuenta con un tamaño de diámetro de 10mm y un grosor de 3.4mm, tamaño similar el nuevo motor elegido según se puede observar en la *Figura 5*.

Teniendo en cuenta que el rango de voltaje de funcionamiento de **B1034.FL45-00-015** abarca un intervalo de 2.3-3.6V en continua, los circuitos diseñados serán equivalentes para este nuevo caso, ya que serán implementados para un voltaje de funcionamiento de 3-3.3V.

En cuanto a la corriente necesaria para el funcionamiento del nuevo dispositivo, según indica el vendedor [12], se recomiendan unos valores de voltaje y tensión de 3V y 180mA, mientras que en **B1034.FL45-00-015** el valor de corriente de funcionamiento propuesto era de 60mA. A pesar de suponer el doble de intensidad no ocasionará dificultades a la hora de desarrollar el producto, estas cuestiones se discutirán en posteriores apartados de este documento relativos a la alimentación y consumo de potencia.

Hemos de añadir que, al tratarse de un motor sin núcleo, obtendremos una respuesta de mayor aceleración en un periodo de tiempo inferior, consiguiendo un arranque y parada más cortos al simular la sensación de inestabilidad tras el disparo. Esto se debe a que los motores sin núcleo poseen una masa menor y según la segunda ley de Newton al disminuir la masa, obtendremos una aceleración mayor.



### 3.1.2 Sensores

#### *Estudio teórico*

Los sensores serán los que aporten los datos físicos al sistema. Recordemos que necesitamos traducir un movimiento en tres dimensiones a una pantalla de dos dimensiones, esto provocará que desde un inicio descartemos la dimensión de la profundidad.

El lugar del puntero en pantalla vendrá definido por una línea recta imaginaria que saldrá del cañón de la pistola hasta su intersección con la pantalla. En caso de salirse de los márgenes, el apuntador se quedará en el borde de estos.

Dado que hemos descartado la dimensión de la profundidad, las únicas dos acciones que podremos llevar a cabo con la pistola en el espacio tridimensional serán:

- La rotación del apuntador en dos ejes.
- El desplazamiento en direcciones coplanarias, tomando como referencia un plano ortogonal al suelo.

Tendremos que ser capaces de discriminar y cuantificar esos movimientos, por lo que, para ello, emplearemos un giroscopio y un acelerómetro. En el caso del primer tipo de movimiento se empleará un sensor giróscopo o giroscopio. Este giroscopio nos permitirá conocer la aceleración angular, de forma que podremos calcular el ángulo de giro que se está efectuando.

En cuanto al segundo desplazamiento utilizaremos un acelerómetro, este permitirá conocer la aceleración en el espacio tridimensional. Así, al combinar ambos movimientos podremos determinar una dirección. El giróscopo es necesario, ya que mide la velocidad angular en caso de que se gire el arma, pero el acelerómetro también, ya que si no se realiza ningún tipo de giro necesitaremos cuantificar el movimiento coplanar a la pantalla.

Con estos dos valores podremos determinar un módulo y una dirección en el espacio, los cuales nos permitirán representar en dos dimensiones la localización del puntero, teóricamente:

$$\omega = \frac{d\theta}{dt} \quad a = \frac{dv}{dt} \quad v = \frac{\Delta s}{\Delta t}$$

$$\text{Calcularemos el módulo según: } v(t) = \int a(t) dt \quad \Delta s = \int_{t_0}^{t_1} v(t) dt$$

Y la dirección en tres dimensiones con:  $\omega = \frac{v \operatorname{sen}(\theta)}{r} \hat{u}$

Por este motivo se ha decidido incorporar en el diseño el sensor **MPU-9250**, el cual es *Multi-Chip Module* (MCM) que cuenta con dos partes integradas en un módulo Quad-Flat No-Leads (QFN) [13]. Una de las partes está formada por el conjunto de un acelerómetro y un giroscopio, de tres ejes cada uno. Por otra parte, tendremos un magnetómetro de tres ejes.

Este pequeño chip consigue integrar los sensores mediante tecnología *Microelectromechanical Systems* (MEMS). Es cierto que el magnetómetro no se ha contemplado en un inicio como un componente necesario, sin embargo, podría servir en un futuro para calibrar el apuntador o añadir mejoras.

Este sensor cuenta con tres *Analog to Digital Converter* (ADC) de 16 bits para cada uno de los sensores internos, lo que nos proporcionará una resolución suficiente en el movimiento a la hora de mapearlo en pantalla. Para añadir precisión en movimientos rápidos y lentos cada una de las partes posee características programables. En referencia a sus fondos de escala tendremos los siguientes rangos [13]:

- Giroscopio: “ $\pm 250, \pm 500, \pm 1000, \text{ and } \pm 2000^\circ/\text{sec}$  (dps)”
- Acelerómetro: “ $\pm 2g, \pm 4g, \pm 8g, \text{ and } \pm 16g$ ”
- Magnetómetro: “ $\pm 4800\mu T$ ”

Para nuestra aplicación particular, en principio, será bastante adecuado, ya que podríamos programarlo con un rango de entre  $\pm 250$  grados por segundo y  $\pm 2g$  midiendo entre -2 y 2 veces la aceleración de la gravedad estándar. Al tener 16 bit en cada ADC para digitalizar las salidas tendremos  $2^{16}$  valores, más que suficientes para poder diferenciar los relativamente pequeños movimientos que se llevarán a cabo. Las tasas de salida máximas del giroscopio y acelerómetro son 8000Hz y 4000Hz respectivamente [13].

El sensor podrá utilizar los protocolos *Inter-Integrated Circuit* (I<sup>2</sup>C) y *Serial Peripheral Interface* (SPI) [13] para comunicarse con componentes externos, por lo que no presentará una interoperabilidad demasiado compleja respecto a la mayoría de microcontroladores. Estos protocolos podrán operar a un máximo de 400kHz y 1MHz [13] respectivamente, motivo que nos incentivará a utilizar SPI.

Finalmente enumerar que su tensión de alimentación vendrá dada entre 2.4-3.6V, con una fuente de alimentación digital separada.

Es necesario explicar que todas estas tasas y posibilidades vendrán determinadas por el conjunto global de elementos y deberemos tener en cuenta varios factores:

En primer lugar, no olvidemos que los sensores únicamente nos proporcionarán datos binarios provenientes de magnitudes físicas, y que nuestro objetivo final es el de representar un movimiento en pantalla proporcional a los movimientos físicos ejercidos sobre nuestro dispositivo. Expuesta la anterior argumentación será necesario entrar a valorar algunas de las características de los dispositivos señaladores.

Una de las características principales de estos dispositivos es la tasa de sondeo o *Polling Rate*, que es la frecuencia con la que el dispositivo apuntador informa de su posición al equipo en el que está conectado. La tasa de sondeo vendrá determinada por las limitaciones del equipo al que se conecte y sus periféricos asociados, de modo que deberán tenerse en cuenta los retardos tales que el retardo de propagación y el retardo del monitor entre otros. De nada nos servirá tener una elevada tasa de sondeo si nuestro monitor no es capaz de representar los datos a esa tasa de *Frames Per Second* (FPS).

Otra de las limitaciones vendrá impuesta por la resolución del monitor en el que se emplee, de manera que deberemos determinar cuántos píxeles por pulgada o *Dot Per Inch* (DPI) establecer en nuestro dispositivo apuntador. Será necesario ajustarlo para que funcione de manera adecuada en la mayoría de monitores. Y es que, en caso de subir los DPI obtendremos una mayor granularidad, ganando precisión, pero también provocaremos que cuando se conecte a monitores de menor resolución sea muy sensible a pequeñas variaciones provocando que su movimiento sea rápido y descontrolado. Algunas de estas cuestiones se tratarán en el apartado de *software*, de cara a la programación del microcontrolador.

En este punto de la argumentación solamente se enumeran los compromisos que tendremos que establecer y se especula sobre algunos de los valores, definiéndolos como suficientes o recomendados, pero que se tendrán que ajustar de manera exacta en los siguientes puntos del proyecto. Únicamente se está tratando de observar a *grosso modo* si estos componentes son viables o es necesario buscar sustitutos para sus funciones.

### ***Implementación práctica***

El conjunto de sensores MEMS corresponderá con el expuesto durante la búsqueda de componentes siendo este el modelo **MPU-9250**. En el caso del conjunto de sensores

tendremos una problemática asociada, y es que, no se han encontrado versiones que cuenten con patillaje, siendo el único modelo la versión QFNL, el cual no cuenta con guías para poder realizar las soldaduras de una forma sencilla. Esto supondrá un motivo de cambio, en el caso de no contar con la instrumentación adecuada como para poder soldar este componente a la *Printed Circuit Board* (PCB).

Como consecuencia, se ha decidido implementar un pequeño circuito integrado que se instalará en nuestra PCB. El circuito integrado elegido cuenta con el propio conjunto de sensores MEMS, así como los condensadores necesarios para su funcionamiento. Cabe destacar que se incluye un condensador de tantalio de 10uF empleado junto con el regulador de tensión [14]. De forma adicional, incluye un regulador de tensión de 5V a 3.3V que facilitará su inclusión en nuestra PCB pudiendo aportarle directamente una alimentación de 5V.

A continuación, se muestra en la siguiente figura una imagen de muestra derivada del anuncio de compra.



**Figura 6.** Conjunto de sensores MEMS MPU-9250 vista preliminar.[14]

### 3.1.3 Microcontrolador

#### *Estudio teórico*

A la hora de elegir un dispositivo electrónico que lleve a cabo el papel de microcontrolador será necesario que cumpla los requisitos suficientes como para poder gestionar el resto de componentes, así como las exigencias *software* marcadas al inicio del estudio del sistema.

De manera que pasaremos a recopilar de forma breve los requisitos que tendrá que superar este integrado:

- Será necesario que sea compatible con USB HID, permitiendo operar tanto con el protocolo USB como con la clase HID de este mismo.
- Una de las características convenientes, sería la de poder operar con la alimentación facilitada por USB, de manera que no fuese necesaria ningún tipo de alimentación externa ni etapas transformadoras de voltaje que ocuparían demasiado tamaño en la PCB.
- Deberá poseer entradas de lectura digitales, con el fin de ser capaz de leer la información de los interruptores de final de carrera, el conmutador y los botones.
- Necesitará contar con salidas digitales de voltaje que permitan controlar la iluminación de los LED, pero también con salidas analógicas y generación de formas de onda *Pulse-Width Modulation* (PWM) para el caso del modo de disparo automático.
- Tendrá que ser compatible con los protocolos de comunicación externa del sensor fijado, ya sea I<sup>2</sup>C o SPI.
- Finalmente, deberá trabajar a la suficiente frecuencia como para que el conjunto del sistema funcione fluido, consiguiendo tasas de transmisión y recepción equiparables a las salidas del sensor y una velocidad de transmisión de datos a través de USB que garantice una mínima tasa de sondeo.

Dicho esto, se expondrá un buen candidato, el microcontrolador **ATmega32U4** de la empresa Atmel y se pasará a enunciar cómo supera estos requisitos, así como las características por las que ha sido seleccionado para ser el centro neurálgico de nuestro producto.

Es cierto que existe una versión bastante similar de este microcontrolador, siendo esta el modelo ATmega16U4, sin embargo, la decisión ha venido enfocada al modelo de mayor memoria *Flash* relativa a *In-System Programming* (ISP), de modo que tendremos mayor cantidad de memoria a la hora de almacenar el código que carguemos en el microcontrolador desde el dispositivo en el que lo programemos. De esta manera contaremos con una memoria de 32Kbytes para el código del programa. Esta decisión se ha tomado con vistas al futuro, dado que, al no conocer aún la cantidad de memoria necesaria, no se ha desarrollado el *software* del producto. La inclinación ha sido hacia una cantidad mayor evitando así problemas de almacenamiento en un futuro y en caso de

ser suficiente, adaptando el producto para nuevas versiones que implementen funciones adicionales.

Centrándonos en su hoja de datos [15] podremos observar que los requisitos anteriormente expuestos son cumplidos en su totalidad, de manera análoga al enunciado de requisitos, podemos exponer:

- Este componente cuenta con una compatibilidad total con la versión 2.0 del protocolo USB (el cual es retrocompatible) "*Complies fully with Universal Serial Bus Specification Rev 2.0*" [15]. Además, la utilización del mismo es programable "*USB Bus Connection/Disconnection on Microcontroller Request*" [15], de modo que nos permitirá programarlo a través de USB sin necesidad de emplear los pines ISP. Al ser compatible totalmente con USB 2.0 podrá emplear la clase HID de USB siendo capaz de utilizarse como dispositivo apuntador o *mouse*.
- La tensión de alimentación del protocolo USB 2.0 estará entre 4.75-5.25V [16], por lo que será válida para alimentar a nuestro microcontrolador que operará a un voltaje de 2.7-5.5V dependiendo de su funcionamiento.
- El chip posee pines configurables como entradas digitales de lectura, las cuales podrían utilizarse para detectar eventos de los actuadores. Además, cuenta con múltiples puertos de este tipo, más que suficiente para obtener información externa de los actuadores. A mayores, todos ellos pueden ser configurados en modo de resistencia interna o *pull-up resistor*, lo que podría facilitar, energéticamente hablando, la lectura del estado de los interruptores de final de carrera.
- En cuanto a la salida de datos, los pines pertenecientes a los grupos de puertos B, C, D, E, y F, según se define en su hoja de datos "*output buffers have symmetrical drive characteristics with both high sink and source capability*" [15], por lo que podremos utilizar estos puertos como puertos de salida para alimentar los diodos emisores de luz. Por otra parte, el chip nos da la oportunidad de generar formas de onda moduladas en anchura PWM, así como elegir las características de los pulsos en los puertos PB5-PB7, PC6-PC7 y PD6-PD7. De modo que estas formas de onda podrán ser utilizadas para controlar el modo de disparo automático del motor solenoide.
- El microcontrolador posee compatibilidad con el protocolo SPI e I<sup>2</sup>C, contando con las conexiones *Serial Clock* (SCLK) en el puerto PB1, *Master Output Slave*

*Input* (MOSI) en el puerto PB2, *Master Input Slave Output* (MISO) en el puerto PB3 y *Select* (SS) en el puerto PB0, permitiendo la configuración Maestro/Esclavo. Es cierto que en la documentación del microcontrolador no se muestra de forma explícita una compatibilidad con I<sup>2</sup>C, sin embargo, se podría implementar mediante SCLK y *Serial Data* (SDA), utilizando el puerto PD1, el cual permite conexión SDA mediante *Two Wire Interface* (TWI), un protocolo creado por la empresa Atmel, propiedad actual de Microchip que se basa en I<sup>2</sup>C. La principal causa que justifica la creación del protocolo TWI es meramente empresarial, de esta manera se evitan problemas legales con la marca registrada I<sup>2</sup>C, propiedad de la empresa Philips, aunque a efectos prácticos TWI es completamente compatible con cualquier elemento que implemente I<sup>2</sup>C. En [17] se especifican las principales diferencias de TWI con I<sup>2</sup>C y se explica su funcionamiento de una manera más detallada.

- Finalmente, el dispositivo soporta tasas de transferencia USB de hasta 1.5Mbits/s que nos permitirán tener un *Polling Rate* adecuado para un dispositivo que actuará como *mouse*.

Es necesario recalcar que la selección de este microcontrolador posee un criterio de fondo que aún no ha sido expuesto. En la práctica existen múltiples microcontroladores que podrían cumplir con los requisitos anteriormente citados, sin embargo, este dispositivo cuenta con un valor añadido que es determinante de cara al desarrollo del producto. Y es que, el microcontrolador ATmega32U4 cuenta con una importante comunidad de usuarios que lo utilizan y respaldan. Su versatilidad y empleabilidad son tales, que incluso iniciativas como Arduino emplean este microcontrolador en sus productos. “*Arduino is an open-source electronics platform based on easy-to-use hardware and software.*” [18], además “*The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.*”[18]. De este modo podremos emplear durante el desarrollo del *software* las herramientas de Arduino tales como su *Integrated Development Environment* (IDE) y su lenguaje de programación, compatible con la familia de microcontroladores AVR de la empresa Atmel. Estas herramientas permitirán un desarrollo temprano, simplificando tareas complicadas como la instalación

del gestor de arranque o *bootloader* y ampliarán el catálogo de bibliotecas *software* disponibles.

### ***Implementación práctica***

De cara a la incorporación de este microcontrolador en la PCB se recurrirá a su inclusión en crudo, es decir, a pesar de que existen integrados de Arduino desarrollados con este microcontrolador que agilizarían el diseño electrónico, se empleará este componente por separado con el fin de optimizar al máximo la utilización de los pines de este. Además, esto aportará un mayor grado de libertad a la hora de tomar decisiones de diseño electrónico. De esta forma obtendremos un diseño específico ajustado a las necesidades electrónicas del producto, pero podremos aprovechar sus facilidades *software* a la hora de utilizar sus controladores o *drivers*, así como la compatibilidad con distintos elementos que aporta su amplia comunidad en lo referente al desarrollo de bibliotecas.

En concreto, el modelo de microcontrolador elegido ha sido el **ATmega32U4-AUR**. En la **Figura 7** se muestra una imagen de su anuncio de venta [19], donde puede apreciarse el patillaje del integrado que facilitará la tarea de soldado. Las cuestiones relativas a su elección serán tratadas durante el desarrollo electrónico esquemático.



**Figura 7.** Anuncio de venta ATmega32U4-AUR, vista preliminar.[19]

## **3.2 Desarrollo electrónico esquemático**

Tras haber esclarecido los componentes principales que formarán parte del circuito, se pasará a tratar las cuestiones técnicas de forma específica en pos de obtener un diseño



electrónico esquemático. Para ello se discutirá, en términos generales, la configuración de las conexiones, así como la alimentación del circuito. Posteriormente se tratará de realizar un diseño esquemático teniendo en cuenta los parámetros técnicos de cada uno de los componentes del circuito. Seguidamente se comprobará la viabilidad del diseño realizado con un análisis de potencia, para concluir con la realización de un esquema relativo al posicionamiento de componentes en la PCB, siendo este el diseño de la placa de circuito impreso.

### **3.2.1 Análisis general de alimentación y conexiones**

Inicialmente se discutirán algunos aspectos generales, referentes al conexionado global del circuito. Para ello se realizará una primera toma de contacto, donde se expondrán de manera esquemática las propuestas iniciales de la creación del conjunto electrónico.

#### **3.2.1.1 Alimentación**

En lo referente a la alimentación, trataremos de concluir cómo será la configuración empleada con la finalidad de determinar las fuentes de corriente y tensión de nuestra PCB.

La propuesta inicial respecto a la alimentación, se basa en la premisa de compatibilidad y sencillez. Expuesta esta idea, se ha tomado la decisión de emplear de forma directa la alimentación de un puerto USB para todo el conjunto electrónico. De esta forma podremos maximizar la compatibilidad, dado que USB se ha convertido en un protocolo de uso masivo, tanto de cara al campo computacional como en otra electrónica de consumo tal que televisiones o transformadores de *smartphones*.

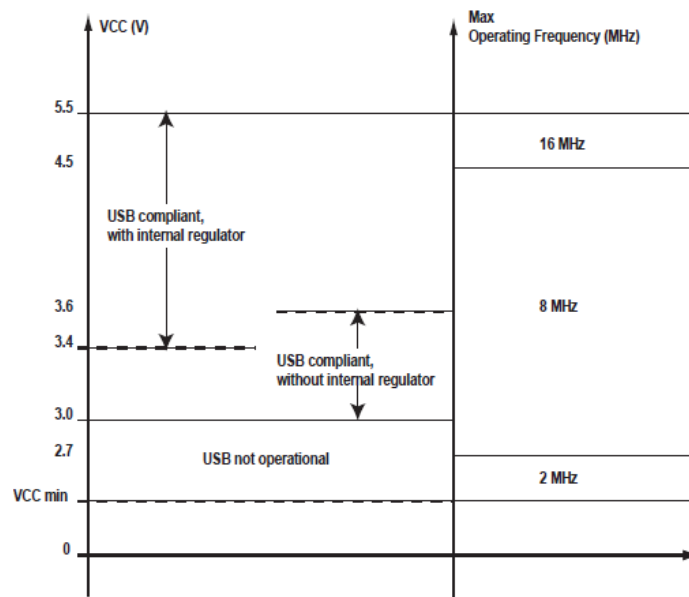
El propósito es utilizar la alimentación del puerto USB 2.0, de modo que se facilitará la compatibilidad con nuevas versiones del protocolo como USB 3.2 e inferiores. Es necesario señalar que en este apartado únicamente se expondrán de forma general los aspectos relacionados con la alimentación, sin embargo, se discutirá en la sección 3.4 de análisis de potencia, las consecuencias de emplear unas u otras versiones del protocolo USB.

De esta manera podremos prescindir de muchos otros elementos que aumentarían el coste y la complejidad del sistema. A pesar de esto, nos encontramos en una fase temprana del

desarrollo y podría haber modificaciones futuras que tratasen de mejorar la calidad del producto de cara a nuevas versiones del mismo.

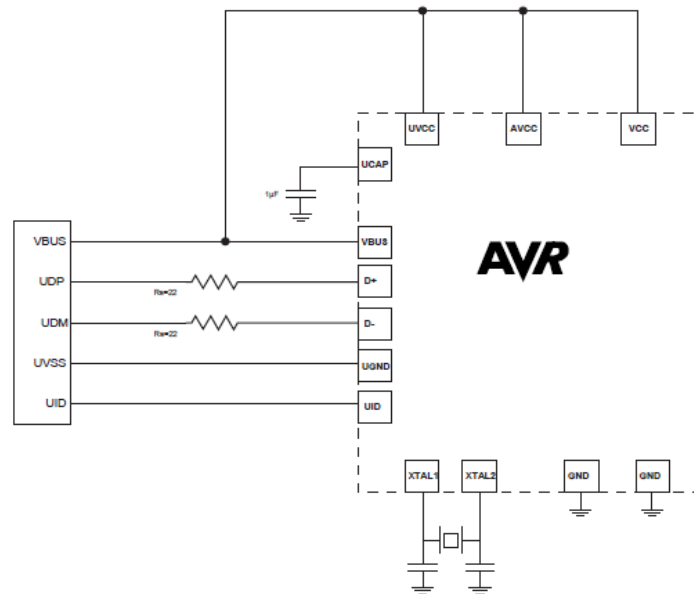
Dado que el principal elemento de nuestro diseño electrónico será el microcontrolador, nos centraremos en estudiar la viabilidad de realizar una implementación mediante alimentación a través de un puerto USB en este elemento.

En la **Figura 8** pueden apreciarse los distintos modos de funcionamiento que se aportan en su hoja de datos [15] dependiendo de la tensión de alimentación aplicada al microcontrolador.



**Figura 8.** *Operating Modes versus Frequency and Power-supply.* [15]

En nuestro caso nos centraremos en el modo de 5V, de manera que el microcontrolador nos permitirá utilizar la máxima frecuencia de trabajo y facilitará la compatibilidad USB. Para ello realizaremos las conexiones típicas que se establecen en su *datasheet* para una alimentación a través del *bus* de alimentación de conexiones USB. El esquema de conexiones elegido viene definido en el documento de características del microcontrolador [15] y puede apreciarse en la **Figura 9**.

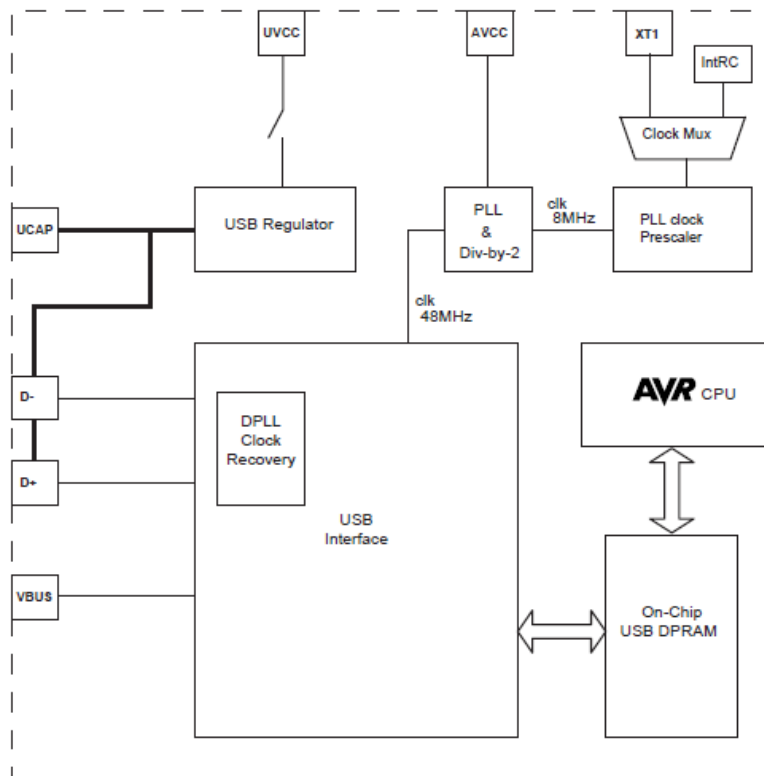


**Figura 9.** Typical Bus Powered Application with 5V I/O. [15]

Si observamos la **Figura 9** veremos cómo el voltaje de alimentación del conector USB está conectado con: el *pad* USB que actúa como entrada al regulador de tensión interno USB (UVCC), la tensión de alimentación analógica (AVCC) y el voltaje en corriente continua (VCC).

También puede verse cómo los pines XTAL2 y XTAL1 están conectados a un oscilador de cristal con dos condensadores a tierra; “Pins XTAL1 and XTAL2 are input and output, respectively, of an inverting amplifier which can be configured for use as an On-chip Oscillator” [15]. Esto se debe a que es necesario incorporar un oscilador externo, ya que el integrado únicamente contará con un oscilador interno de 8MHz. Como consecuencia, para poder aprovechar la frecuencia de trabajo del modo de 5V será necesario introducir un oscilador de cristal de cuarzo externo o un resonador cerámico. En el caso de introducir un cristal con una frecuencia de trabajo de 16MHz los valores recomendados de los condensadores serán de 22pF para ambos.

Finalmente hay que comentar que, será necesario introducir un condensador de 1µF en el pin UCAP para un funcionamiento correcto del regulador de voltaje interno, el cual nos permitirá tener salidas de 3.3V para su empleo en USB. Como aclaración final a la configuración de los pines de salida para el modo de 5V y la alimentación a través del *bus* de entrada USB se adjunta un diagrama de bloques para esta distribución, el cual puede apreciarse en la **Figura 10**.



**Figura 10.** *USB controller Block Diagram Overview.* [15]

### 3.2.1.3 Conexiones

Tras fijar la alimentación del dispositivo principal, será necesario tomar partida en lo referente a la alimentación de los componentes secundarios, así como sus conexiones para interactuar con el resto del circuito. En principio, los componentes externos vendrán alimentados por los pines de salida del microcontrolador, a excepción de los motores. En este apartado pasarán a describirse los procedimientos empleados para la conexión de los distintos componentes externos que compondrán el conjunto del sistema.

En las características de corriente continua o *Direct Current* (DC) del microcontrolador se especifica que las salidas de alto voltaje podrán contar con un mínimo de 4.2V (para su lectura como valor de tensión en alta) siempre que la tensión de alimentación en continua VCC sea de 5V y la salida en modo de alta intensidad sea de 10mA [15].

Según figura en la hoja de datos del microcontrolador “*each I/O port can source more than the test conditions (20mA at VCC = 5V)*” [15]. Aunque se especifica que la suma de todas las salidas de alta intensidad (IOH) en ciertos puertos no deberían exceder los 100mA, por lo que si conectamos muchos de estos pines como salida será necesario tener en cuenta que quizás no se alcance el amperaje deseado. Estas cuestiones relacionadas

con la potencia serán tratadas en secciones posteriores de este documento. Dicho esto, pasaremos a estudiar la configuración para los distintos componentes del producto final.

### ***LED***

Para los LED podremos utilizar cualquiera de los pines de salida digitales del microcontrolador con el fin de controlar el LED y a la vez alimentarlo. Recordemos que el voltaje en directa es  $V_F \sim 1.8V$  y que los valores típicos de intensidad luminosa venían dados para fotocorrientes de 20mA, por lo que la salida de los pines digitales será suficiente como para cumplir este objetivo. Aunque se hará necesario incluir resistencias para limitar la corriente salida a un máximo de 20mA.

Hay que destacar, que según se expuso en la sección de búsqueda de componentes, se incorporarán tres diodos emisores de luz, con la finalidad de poder abarcar un ancho de haz equivalente a una emisión de luz cuasi isotrópica.

### ***Interruptores***

En el caso de los interruptores de final de carrera no será necesario ningún tipo de alimentación, ya que únicamente funcionarán estableciendo una conexión entre los pines de tierra y los pines de lectura digital del microcontrolador. Para ello se configurarán los pines de lectura del microcontrolador en el modo *pull up*. De esta forma se podrá detectar cuando han sido pulsado los mismos, dado que si la tensión de entrada en ese pin corresponde con la tensión de referencia de tierra significará que el interruptor ha sido accionado. En el caso contrario el pin tendrá una tensión que vendrá determinada por el voltaje que cae en el modo de configuración *pull up* (correspondiente con un valor de tensión que se leerá como tensión en alta).

Como valor añadido a la búsqueda de componentes, se incorporará un interruptor genérico que permitirá seleccionar el modo de operación, pasando de un modo de disparo básico, al modo de disparo automático.

### ***Display LED***

Como adición al conjunto de elementos expuestos durante la búsqueda de componentes se incorpora un *display* LED de 2 dígitos que permitirá seleccionar al usuario múltiples configuraciones y parámetros, visualizando los cambios efectuados en dicha pantalla. Para poder llevar a cabo la incorporación del selector de balas de la recámara será

necesario que el usuario reciba una realimentación sobre el estado actual de la capacidad de balas del cargador. Para ello se implementará un *display* LED de 2 dígitos, que permitirá seleccionar el tamaño del cargador. En nuestro caso, emplearemos un *display* de ánodo común con el fin de intentar reducir el consumo del dispositivo, permitiéndonos operar con ceros lógicos para su activación o entradas en baja, las cuales necesitarán una menor potencia para desempeñar la misma función.

### ***Botones***

Hasta ahora estos elementos no se habían nombrado en el transcurso del documento, sin embargo, se hace necesario incluirlos en este punto, dado que tratamos de aportar un mayor grado de detalle en la construcción de las conexiones.

Durante la búsqueda de opciones de compra de los componentes se originó la idea de poder incluir diversos botones en la construcción del producto. De esta manera sería más fácil adaptar y compatibilizar la integración de la pistola en diversos videojuegos.

La necesidad inicial de incorporación de botones ha venido marcada por la incorporación del selector de balas del cargador y por el hecho de que estos productos suelen venderse en lotes que incorporan varios botones a un bajo coste. Con todo esto se ha determinado que sería interesante incorporar botones en el perfil anterior de la carcasa. La incorporación de estos botones no tendría otra función que aportar compatibilidad, de manera que se pudiera jugar con ella en juegos que precisaran otras acciones como saltar o agacharse, sin tener que emplear otro control adicional.

En lo referente a sus conexiones se procederá de una manera idéntica a la expuesta para los interruptores de final de carrera.

### ***Motores***

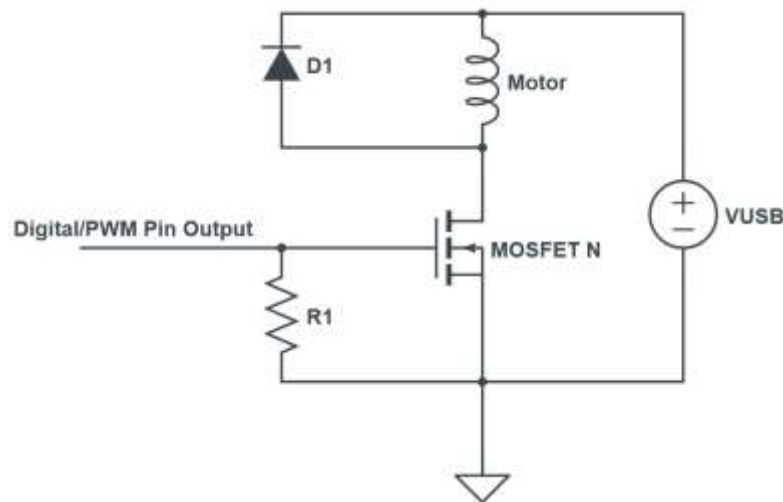
En el caso de la configuración de conexiones de los motores necesitaremos utilizar un pin del microcontrolador que pueda actuar como pin digital y proporcione estos valores de tensión y corriente, pero que además cuente con la posibilidad de generar un tren de pulsos modulado con PWM con el fin de implementar el disparo automático.

Según lo especificado en la hoja de datos del microcontrolador no será posible proporcionar la intensidad máxima de salida en los pines digitales o PWM requerida por

los motores, como consecuencia será necesario emplear algún tipo de configuración electrónica a mayores.

La utilización de motores a nivel electrónico es similar al empleo de bobinas, dado que ambos elementos se basan en inducción magnética para producir movimientos físicos. Como consecuencia, se hará indispensable la incorporación de circuitos de seguridad que permitan garantizar que no se producen corrientes indeseadas, las cuales podrían dañar otras partes del circuito y que permitan, a su vez, aportar la potencia necesaria para la actuación de estos elementos.

Una de las opciones podría ser la representada en la **Figura 11**. En esta, puede apreciarse que se ha incorporado un diodo entre la tensión de alimentación del solenoide y el resto de componentes. La inclusión del diodo permitiría evitar que la tensión generada por el solenoide cuando este deja de actuarse volviera hacia el interior del microcontrolador. Por otro lado, se ha situado un transistor MOSFET de tipo N, el cual actuará como un interruptor conectando el solenoide con tierra cuando la señal llegue por el pin correspondiente del microcontrolador.



**Figura 11.** Implementación para control de motor

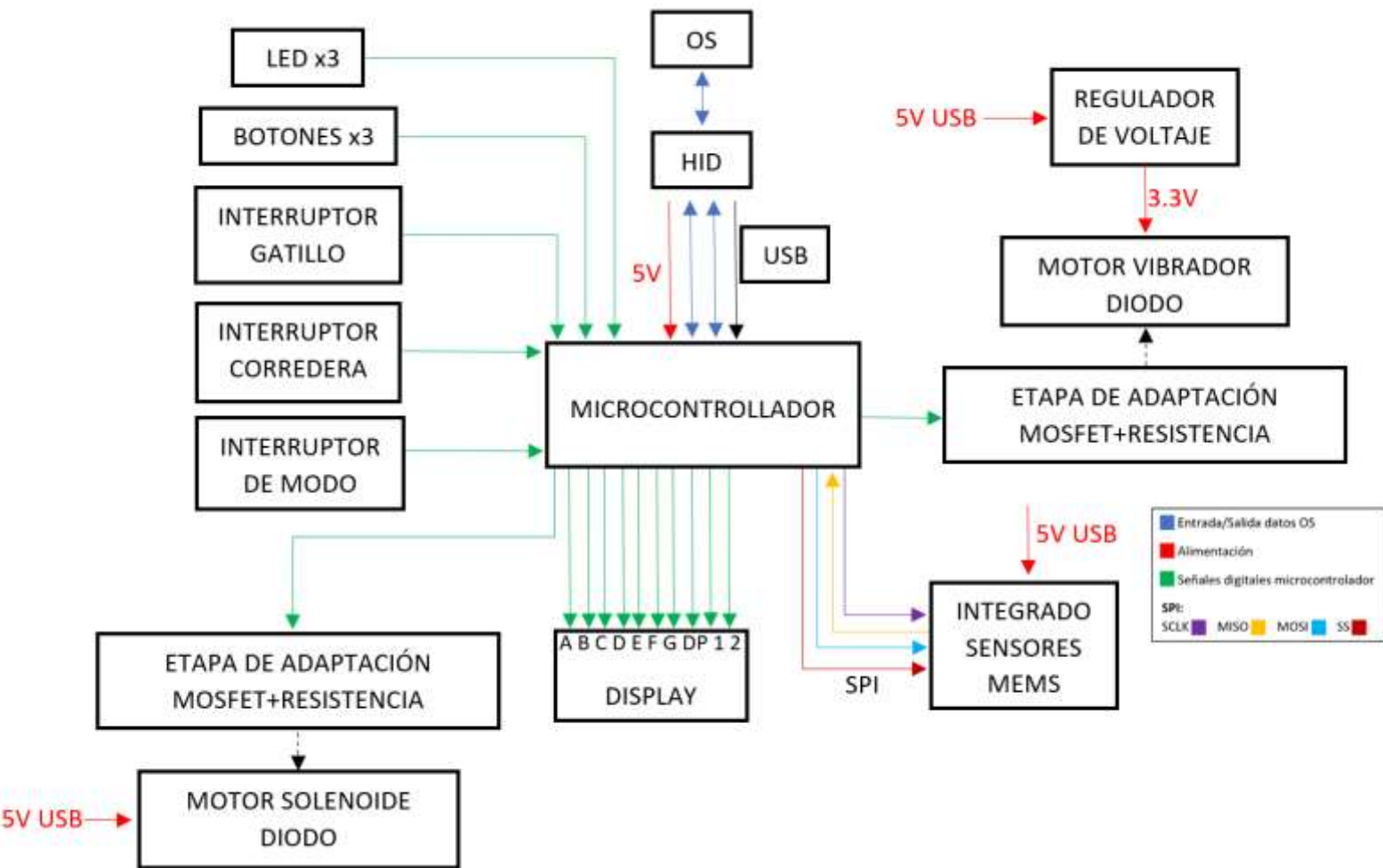
De esta forma, cuando la señal proveniente del microcontrolador esté desactivada, el transistor no permitirá el paso de corriente entre drenador y fuente. Por el contrario, cuando la señal del pin del microcontrolador esté activada, el transistor pasará a la zona de saturación y, a efectos prácticos, será como tener activado el motor. Esta podría ser una manera práctica de conectar los motores, ya que podríamos conectar la alimentación de los motores directamente a la fuente de alimentación externa USB. Esta configuración circuital será tratada durante el desarrollo esquemático de forma más exhaustiva.

Cabe destacar que el motor vibrador poseía una tensión de alimentación en continua de 2.3-3.6V. El microcontrolador seleccionado no posee salidas de ese rango de voltaje y solo cuenta con un regulador de tensión interno para las salidas USB, con lo cual será necesario incorporar un regulador de tensión con entrada a 5V y salida a 3.3V.

### Sensores

En cuanto a los sensores nos encontraremos en el mismo caso que en el del motor vibrador. La tensión de alimentación de los sensores integrados será de 2.4-3.6V, sin embargo, el circuito integrado empleado ya incorpora un regulador de tensión y podremos conectarlo directamente a 5V. A mayores, será necesario establecer las conexiones del protocolo SPI utilizado.

Tras haber establecido los componentes principales del sistema, se muestra en la **Figura 12** un esquema conceptual que engloba el resumen de conexiones discutidas durante este apartado.



**Figura 12.** Esquema conceptual de carácter general con las conexiones del circuito.



Entre estos elementos se obvian algunos de los componentes que conforman la posible implementación final como conectores, osciladores, condensadores y resistencias, que serán analizados en los siguientes apartados del proyecto.

### **3.2.2 Implementación del desarrollo electrónico esquemático**

La realización de los esquemáticos se ha llevado a cabo con el programa Proteus [20] en su versión 8.9, que es la que facilita una mejor compatibilidad con las bibliotecas de componentes externas de SamacSys [21]. Permitiendo la inclusión de estos desde la propia interfaz del programa y evitando el empleo de programas externos como puede ser la herramienta escrita en Python, Library Loader [22].

Una vez establecido un primer contacto con el programa Proteus 8.9 y esclarecido el empleo de bibliotecas externas con el fin de llevar a cabo la realización de los esquemáticos del proyecto, pasaremos a ver cómo se han implementado las distintas conexiones del conjunto de elementos electrónicos.

A lo largo de esta sección se tratarán las conexiones desde el punto de vista de sus interacciones frente al microcontrolador, dado que será este el encargado de controlar el funcionamiento global del circuito. Así mismo, se aportarán detalles técnicos específicos de cada una de las secciones asociadas con los componentes principales y los elementos electrónicos que los conforman. También se arrojará luz sobre la configuración de las distintas huellas asociadas a los componentes, así como aquellas que se han creado de manera específica en el programa Proteus 8.9.

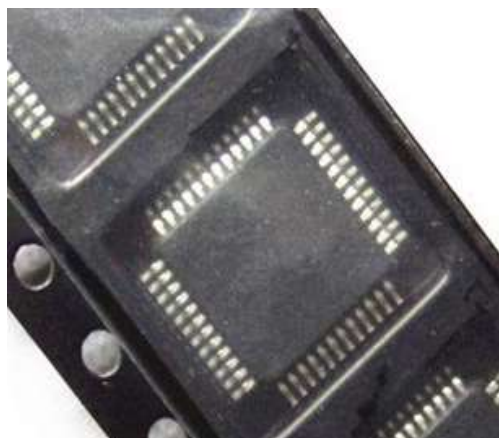
En primer lugar, se tratará de esclarecer el modelo concreto de microcontrolador, una vez definido el componente genérico con anterioridad. Específicamente se llevará a cabo la implementación de su modelo ATmega32U4-AU o ATmega32U4-MU ya que son los que nos permitirán emplear cristales de cuarzo externos en lugar de cavidades resonantes *Resistance-Capacitor* (RC). Tal y como se define en su hoja de datos, se muestran las variaciones de sus distintos modelos en la *Figura 13*.

Speed [MHz]	Power Supply	Ordering Code	Default Oscillator	Package	Operation Range
16	2.7 - 5.5V	ATmega32U4-AU	External XTAL	44ML	Industrial (-40° to +85°C)
		ATmega32U4RC-AU	Internal Calib. RC		
		ATmega32U4-MU <sup>(1)(2)(3)</sup>	External XTAL	44PW	
		ATmega32U4RC-MU <sup>(1)(2)(3)</sup>	Internal Calib. RC		

Notes: 1. For more information on running the USB from internal RC oscillator consult application note AVR291: 8MHz Internal Oscillator Calibration for USB Low Speed on Atmel ATmega32U4RC.  
2. USB operation from internal RC oscillator is only guaranteed for 0°C to 40°C.  
3. These parts are shipped with no USB bootloader pre-programmed.

**Figura 13.** Distintas variantes del microcontrolador ATmega32U4. [15]

Nos decantaremos por el modelo ATmega32U4-AU dado que no se ha encontrado una versión *Thin Quad Flat Package* (TQFP) de la variante “MU”. De esta forma se facilitará la tarea del soldado de componentes en la PCB, debido al patillaje del integrado. A mayores, en el mercado de componentes podremos encontrarnos distintas variantes de este chip, referidas al encapsulado del paquete. Una de estas variantes, la cual será seleccionada para el proyecto, será la variante ATmega32U4-AUR, donde la letra “R” indica *Tape and Reel* que es la forma en la que viene ensamblado el dispositivo. Puede observarse el empaquetado en la **Figura 14**.



**Figura 14.** Empaquetado del chip ATmega32U4-AUR

Al tratar de importar el microcontrolador ATmega32U4, las bibliotecas de Proteus no contenían el modelo apropiado, por lo que ha sido necesario emplear modelos *Electrical Computer Aided Design* (ECAD) provenientes de fuentes externas. Para el desarrollo del esquemático del proyecto se han empleado principalmente diseños obtenidos a través del motor de búsqueda de componentes SamacSys [23].

### 3.2.2.1 Configuración de las conexiones principales del microcontrolador

#### *Entradas Digitales*

Tal y como figura en la hoja de datos del microcontrolador, los puertos B, C, D, E, y F pueden actuar como entradas/salidas (E/S) digitales. Además, poseerán la capacidad de configurarse en modo *pull-up*, activándose mediante *software*.

Cada puerto estará formado por tres direcciones de memoria que contendrán el puerto al que pertenece (PORTx), la dirección de memoria en la que se encuentra el puerto (DDRx), y el número de pin del puerto (PINx). Se adjunta en la **Figura 15** una pequeña aclaración visual procedente de la hoja de datos del microcontrolador que permite comprender el funcionamiento de los puertos.

#### 10.4.2 Port B Data Register – PORTB

Bit	7	6	5	4	3	2	1	0	
	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	PORTB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

#### 10.4.3 Port B Data Direction Register – DDRB

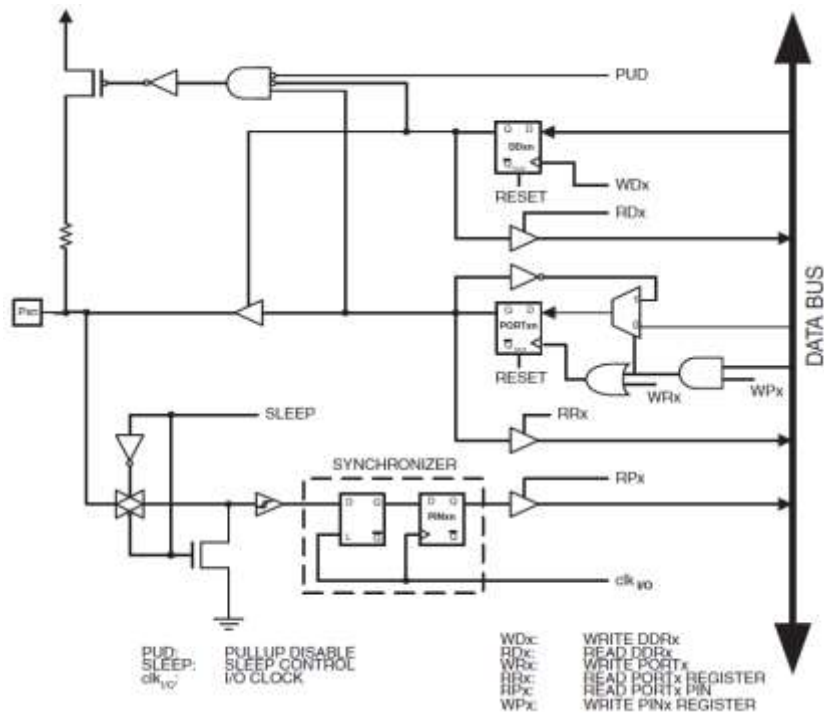
Bit	7	6	5	4	3	2	1	0	
	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	DDRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

#### 10.4.4 Port B Input Pins Address – PINB

Bit	7	6	5	4	3	2	1	0	
	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	PINB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

**Figura 15.** Esquema representativo sobre las direcciones y registros de los puertos B del microcontrolador empleado. [15]

De esta manera, podrán utilizarse de forma lógica como pines digitales E/S. El esquema general de la entrada y salida de datos para los pines digitales del microcontrolador se muestra en la **Figura 16**. Es por esto que, en nuestro diseño, utilizaremos principalmente todos los pines de los puertos B, C, D, E, y F por defecto como pines de E/S digitales, salvo en algunos casos puntuales en los que utilizaremos las funciones alternativas de los mismos.



**Figura 16.** Esquema circuital representativo sobre el funcionamiento de los pines digitales en el microcontrolador. [15]

### Funciones alternativas de los puertos

Para el resto de los pines que no estarán efectuando su rol de E/S digital, se emplearán algunas de sus funciones alternativas.

En primer lugar, será necesario utilizar los puertos SPI para poder establecer la comunicación con el sensor MPU-9250. Con lo que se utilizarán las funciones alternativas de los puertos PB0 (SS), PB1 (SCLK), PB2 (MOSI) y PB3 (MISO), cuya descripción de la hoja de datos puede observarse en la **Figura 17**.

PB3	PDO/MISO/PCINT3 (Programming Data Output or SPI Bus Master Input/Slave Output or Pin Change Interrupt 3)
PB2	PDI/MOSI/PCINT2 (Programming Data Input or SPI Bus Master Output/Slave Input or Pin Change Interrupt 2)
PB1	SCK/PCINT1 (SPI Bus Serial Clock or Pin Change Interrupt 1)
PB0	$\overline{SS}$ /PCINT0 (SPI Slave Select input or Pin Change Interrupt 0)

**Figura 17.** Funciones alternativas de los puertos PB0-3. [15]

También se emplearán funciones alternativas en el caso de las señales de control de los motores, tanto el motor vibrador, como el solenoide. Para ello se utilizarán señales con modulación de ancho de pulso PWM. Múltiples pines poseen esta característica

alternativa. En nuestro caso emplearemos los pines PC6 y PB6. Las descripciones de estos pines pueden apreciarse en la **Figura 18**.

PC6	OC.3A/OC4A (Output Compare and PWM output A for Timer/Counter3 or Output Compare and complementary PWM output $\bar{A}$ for Timer 4)
PB6	OC1B/PCINT6/OC.4B/ADC13 (Output Compare and PWM Output B for Timer/Counter1 or Pin Change Interrupt 6 or Timer 4 Output Compare B / PWM output or Analog to Digital Converter channel 13)

**Figura 18.** Funciones alternativas de los pines que emplearán PWM. [15]

### **Conexiones USB**

Tal y como se ilustró en la sección 3.2.1.1, la alimentación del sistema vendrá implementada a través de los puertos destinados a USB. Por lo que, continuando con las configuraciones indicadas en la hoja de datos se conectarán las entradas de voltaje con la de VBUS. Y siguiendo las recomendaciones aportadas en su hoja de datos, se incorporarán dos resistencias de 22 ohmios en los pines de datos USB. El esquema de estas conexiones se observa en la previamente expuesta **Figura 9**.

Cabe recalcar que los elementos pasivos incorporados en el esquemático desarrollado mediante el programa Proteus, tales que resistencias y condensadores, se han seleccionado sin tener en cuenta el modelo, empleando para ello componentes genéricos con una tolerancia lo más pequeña posible en relación al criterio calidad-precio.

Posteriormente se ha incluido un conector de mini USB conectando el pin ID del mismo con la toma de tierra. Esta conexión se ha efectuado dado que se conectará a un puerto USB de 4 pines en nuestro equipo, y no de 5. El argumento a favor de la utilización del conector mini USB tipo B es su pequeño tamaño y amplio uso, de modo que facilitará la incorporación del conector del cable dentro de la carcasa de la pistola. Además según se especifica en [16] “*The physical difference in the Series “A” and “B” connectors insures proper end user connectivity. The “A” connector is the principle means of connecting USB devices directly to a host or to the downstream port of a hub.*” y “*The “B” connector allows device vendors to provide a standard detachable cable*”, de modo que, con el objetivo de utilizar las recomendaciones de la especificación USB 2.0 utilizaremos el conector tipo A en el lado del equipo en el que se conecte nuestro producto y tipo B a la entrada del circuito de nuestra pistola.

En cuanto al pin AREF, según se especifica en su hoja de datos, estará destinado a ser la referencia del conversor analógico digital “*This is the analog reference pin (input) for the A/D Converter.*”[15]. Se especifica, también en la hoja de datos del microcontrolador, como al estar conectado al ADC, se mejorará la precisión de este si se añade un condensador de desacoplo que permita “filtrar” el ruido de la señal de continua. Recordemos que la impedancia teórica de un condensador vendrá dada por la relación matemática:

$$Z = \frac{1}{j\omega C}$$

Por lo que, al llegar una señal ruidosa, el condensador únicamente dejará pasar a través de él esta señal, concebida como una posible señal de corriente alterna. Por lo tanto, situando este condensador en paralelo, se conseguirá que la señal ruidosa fluya a través de este y supondrá una impedancia idealmente infinita que facilitará el transcurso de la señal en corriente continua al ADC. Sin embargo, el valor del condensador no se especifica en la hoja de datos.

Si observamos las recomendaciones prácticas del fabricante a la hora de implementar sus integrados en circuitos electrónicos se cita:

*“To improve the accuracy of the ADC, the analog supply voltage must be decoupled separately, similar to the digital supply voltage. AREF must also be decoupled. The typical value for the capacitor is 100nF. If a separate analog ground (AGND) is present, the analog ground should be separated from the digital ground so that the analog and digital grounds are only connected at a single point (at the power supply GND).”* [24].

Por este motivo, en nuestro diseño, implementaremos el valor típico citado en el documento de consideraciones de diseño de la empresa que fabrica el microcontrolador, siendo este de 100nF. Además, se incorporarán condensadores de esta capacidad en todos los puntos del circuito que transporten valores de alimentación con el fin de mejorar la *Signal to Noise Ratio* (SNR).

Como podemos apreciar, esto es coherente con el condensador añadido en el puerto UCAP, el cual tenía un valor algo mayor, siendo de 1μF. En posteriores versiones del producto se podría barajar la opción de implementar ambos condensadores de 1μF, con el fin de facilitar la implementación y obtención de materiales.

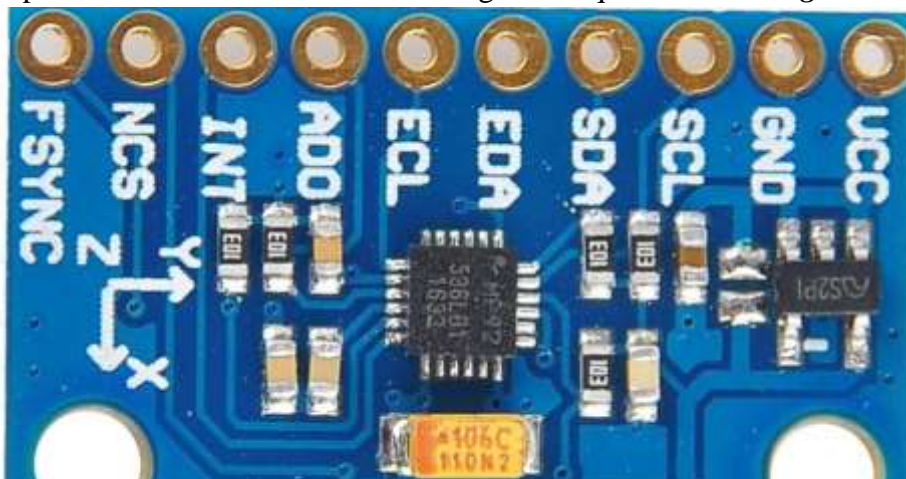
Finalmente, para terminar con las conexiones necesarias para el funcionamiento del microcontrolador (sin contar con sensores ni actuadores), se indica, en la hoja de datos del mismo, que los pines que estén sin usar será necesario configurarlos en modo *pull-up* utilizando la “resistencia interna” del microcontrolador. Según se cita: “*If some pins are unused, it is recommended to ensure that these pins have a defined level.*”, “*The simplest method to ensure a defined level of an unused pin, is to enable the internal pull-up*”, “*Connecting unused pins directly to VCC or GND is not recommended, since this may cause excessive currents if the pin is accidentally configured as an output.*” [15].

En la práctica, para este caso y el resto de conexiones del circuito que no lleguen a emplearse, se dejarán sin conectar con ningún otro punto con el propósito de evitar cortocircuitos o flujos de corriente indeseados que puedan dañar los componentes de la PCB.

### 3.2.2.2 Conexiones con sensores

Tal y como se expuso durante la búsqueda de componentes, en el caso de los sensores emplearemos un integrado ya fabricado que se conectará como un gorro o *hat* a nuestro circuito, la utilización de este elemento viene determinada por la complejidad de realizar las soldaduras del chip integrado MPU-9250.

Sin embargo, se realizará un pequeño estudio, con el objetivo de comprobar que la configuración circuital ofrecida por el fabricante del módulo integrado con el conjunto de sensores MEMS MPU-9250 es correcta. Para ello, nos centraremos en las recomendaciones de implementación del sensor mostradas en su hoja de datos, las cuales serán comparadas con la construcción del integrado adquirido. En la **Figura 19** podemos



**Figura 19.** Visualización de alta calidad del conjunto integrado MPU-9250 implementado en una PCB.

observar una imagen ampliada del conjunto integrado adquirido, en la que se visualizan de manera más detallada los componentes del mismo.

Tal y como se muestra en la imagen de la **Figura 19**, nos encontramos ante una implementación que además de añadir el conjunto de sensores MEMS, aporta un regulador de tensión, condensadores y resistencias.

En la figura anterior podemos apreciar a primera vista cómo el regulador de tensión se encuentra conectado directamente con el pin VCC y con la entrada de habilitación del regulador, además en este punto cuenta con un condensador. A la salida del mismo poseemos una de las patillas (*bypass*) sin conectar. En cuanto a la otra, que es la salida, está conectada en paralelo con dos condensadores, uno mostrado en claro, y otro correspondiente con el condensador nombrado como “106C110N2” que es un condensador electrolítico de tántalo, de ahí su gran tamaño. La utilización de condensadores de este tipo a la salida de los reguladores de tensión no suele ser muy habitual, debido al gran tamaño que ocupan en el circuito, sin embargo, sí que suele ser recomendada por los fabricantes como consecuencia de su baja *Equivalent Series Resistance* (ESR) que mejorará la respuesta transitoria del circuito, acelerando el

Pin Number	Pin Name	Pin Description
1	RESV	Reserved. Connect to VDDIO.
7	AUX_CL	I <sup>2</sup> C Master serial clock, for connecting to external sensors
8	VDDIO	Digital I/O supply voltage
9	AD0 / SDO	I <sup>2</sup> C Slave Address LSB (AD0); SPI serial data output (SDO)
10	REGOUT	Regulator filter capacitor connection
11	FSYNC	Frame synchronization digital input. Connect to GND if unused.
12	INT	Interrupt digital output (totem pole or open-drain)
13	VDD	Power supply voltage and Digital I/O supply voltage
18	GND	Power supply ground
19	RESV	Reserved. Do not connect.
20	RESV	Reserved. Connect to GND.
21	AUX_DA	I <sup>2</sup> C master serial data, for connecting to external sensors
22	nCS	Chip select (SPI mode only)
23	SCL / SCLK	I <sup>2</sup> C serial clock (SCL); SPI serial clock (SCLK)
24	SDA / SDI	I <sup>2</sup> C serial data (SDA); SPI serial data input (SDI)
2 – 6, 14 - 17	NC	Not internally connected. May be used for PCB trace routing.

**Figura 20.** Tabla resumen acerca de las funcionalidades de los pines del sensor MPU-9250.[13]



amortiguamiento de la señal. Esta hipótesis podrá comprobarse cuando tratemos de esclarecer la implementación del regulador de tensión del motor vibrador.

Dejando atrás la configuración del regulador de tensión, nos centraremos en las conexiones del módulo MPU-9250. En la **Figura 20** se resumen las funcionalidades de cada uno de sus pines de salida.

Como podemos observar en la figura anterior, los pines NC no estarán conectados al circuito interno que implementa los tres sensores, por esto se dejarán sin conectar.

Acto seguido nos centraremos en los pines RESV, los cuales están reservados a funciones específicas. Uno de ellos irá conectado con la toma inicial de alimentación, el segundo pin RESV se dejará con la misma configuración que los pines NC (pin 19) y finalmente el último pin RESV irá conectado a la toma de tierra del circuito (pin 20).

En lo referente a la alimentación, se empleará la misma entrada (salida del regulador de voltaje) tanto para la fuente de alimentación digital del conjunto MEMS (VDDIO), como para la general (VDD). Ambos estarán conectados con el pin RESV (pin 1).

En lo relativo a las entradas con conexiones a tensiones de alimentación, se indica en la hoja de datos que para un funcionamiento correcto se deberían incluir los condensadores mostrados en la **Figura 21**.

Component	Label	Specification	Quantity
Regulator Filter Capacitor	C1	Ceramic, X7R, 0.1 $\mu$ F $\pm$ 10%, 2V	1
VDD Bypass Capacitor	C2	Ceramic, X7R, 0.1 $\mu$ F $\pm$ 10%, 4V	1
VDDIO Bypass Capacitor	C3	Ceramic, X7R, 10nF $\pm$ 10%, 4V	1

**Figura 21.** Condensadores propuestos en la hoja de datos del conjunto de sensores MEMS. [13]

Siguiendo con las conexiones de los pines del conjunto de sensores, nos centraremos ahora en los pines AUX\_DA y AUX\_CL. Estas conexiones están orientadas a I<sup>2</sup>C, por el contrario, nosotros utilizaremos el protocolo SPI debido a sus características *full-duplex* y a sus altas tasas de transferencia.

Es cierto que sería posible implementar conexiones mediante I<sup>2</sup>C (TWI) ya que, a pesar de que en la hoja de datos del microcontrolador no figure su compatibilidad de forma explícita, el integrado ATmega32u4 posee los pines SDA y SCL que permitirían la compatibilidad de comunicación con los sensores mediante I<sup>2</sup>C (TWI). Como en nuestro proyecto desconocemos aún la tasa de transferencia necesaria para la comunicación con

el sensor y en vista a desarrollar un producto escalable, nos decantaremos por el empleo del protocolo SPI. El único inconveniente de SPI sería su falta de control de flujo y la necesidad de utilizar cuatro de los pines del microcontrolador en vez de dos, sin embargo, esto no será un problema de cara a la implementación del producto. Dicho esto, los pines AUX\_DA y AUX\_CL quedarán sin utilizar.

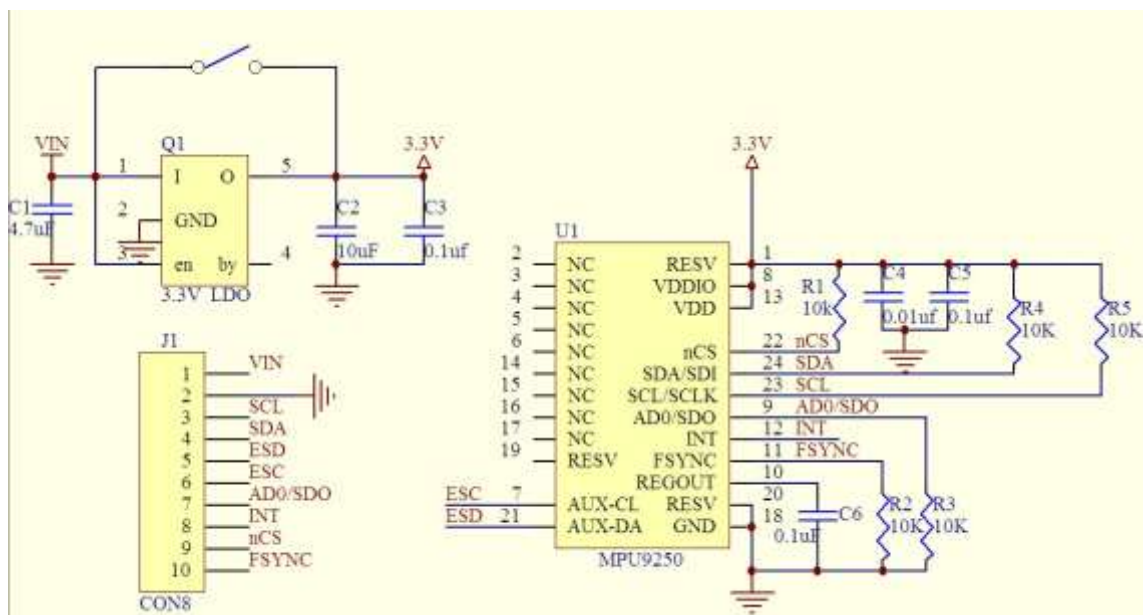
El pin FSYNC tampoco se empleará, por lo que quedará también sin conectar. En la hoja de datos no se especifica mucho sobre este pin, sin embargo, parece que podría emplearse para guardar las mediciones de los sensores en registros y esto, en nuestro caso, no será necesario, ya que emplearemos las mediciones de los sensores en tiempo real.

Siguiendo con el estudio del patillaje del MPU-9250 nos centraremos en el pin INT. Este pin se podrá utilizar para gestionar interrupciones. Según se muestra en su hoja de datos se podrán gestionar 4 tipos de interrupciones "*Items that can trigger an interrupt are (1) Clock generator locked to new reference oscillator (used when switching clock sources); (2) new data is available to be read (from the FIFO and Data registers); (3) accelerometer event interrupts; and (4) the MPU-9250 did not receive an acknowledge from an auxiliary sensor on the secondary I2C bus. The interrupt status can be read from the Interrupt Status register.*" [13]. Dado que nosotros trabajaremos con un sistema en tiempo real y nuestro producto deberá funcionar de manera instantánea al conectarse, no emplearemos ninguna de las posibles interrupciones que se pueden implementar. Por este motivo el pin INT quedará sin conectar.

Finalmente, solo nos quedarán por conectar los 4 pines correspondientes con las conexiones SPI. El pin nCS servirá para seleccionar el chip activo de la conexión SPI, dado que este protocolo permite múltiples conexiones con distintos chips. Según se cita en su hoja de datos "*The CS lines of the non-selected slave devices are held high, causing their SDO lines to remain in a high-impedance (high-z) state so that they do not interfere with any active devices.*" [13], con lo cual, se podrá implementar esta conexión a pesar de contar únicamente con un dispositivo SPI conectado. Esta decisión viene propiciada principalmente con aspectos *software*, que serán expuestos el en anexo, junto al resto de código del programa.

El pin SCL irá conectado a la comunicación serie de la señal de reloj del microcontrolador (SCLK). Y en cuanto a los pines SDI, entrada de datos al sensor y SDO, salida de datos del sensor, se conectarán respectivamente a los pines MOSI y MISO del microcontrolador.

Una vez estudiada la configuración del integrado adquirido, las recomendaciones de conexión del fabricante y las conexiones que se aportarán desde nuestro circuito con este integrado, se aporta en la **Figura 22** un diseño esquemático proporcionado por un vendedor que pone a nuestro disposición el mismo producto adquirido.



**Figura 22.** Diseño esquemático aportado por un vendedor independiente sobre la PCB que incorpora el conjunto de sensores MPU-9250.

Como podemos observar, no se incorpora el condensador VDDIO recomendado por el fabricante y se añaden algunas resistencias a mayores. Sin embargo, la configuración circuital del conjunto integrado es correcta.

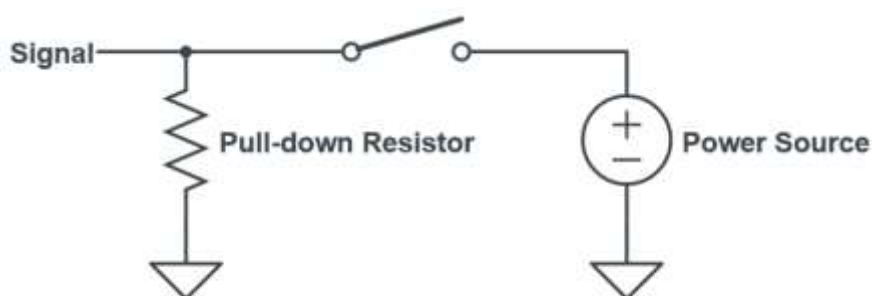
De cara a nuestro diseño únicamente se emplearán los pines VCC, GND, SCL, SDA, ADO y NCS de la PCB adquirida. A la hora de incorporar este componente a nuestro diseño esquemático ha sido necesario crear un nuevo diseño desde cero, dado que no se han encontrado bibliotecas que contengan la huella y modelo de este elemento. Es por esto que, se ha creado un elemento definido en el diseño esquemático global como U2 al que se le ha asignado una huella CON10\_1x10\_HDR\_2663. Esta huella se basa principalmente en un conector de 10 pines de un tamaño algo inferior al que posee este integrado, pero que nos facilitará una huella más pequeña y permitirá el desarrollo de una PCB de menor tamaño, con el objetivo de que nuestro producto no posea grandes dimensiones.

### 3.2.2.3 Conexiones con motores

A la hora de realizar las conexiones con los diferentes motores de nuestro circuito será necesario incorporar una circuitería extra que nos permita proteger el conjunto de componentes, así como aportar fiabilidad al sistema global. Por lo que, en los siguientes apartados, se tratará de argumentar la elección de los componentes del circuito de adaptación de los motores, así como las aportaciones al mismo y las mejoras incorporadas.

#### *Resistencia*

En primer lugar, se aportarán algunas de las bases teóricas en relación a la resistencia con la que contará esta etapa circuital. Se empleará una resistencia que actuará con una configuración *pull-down*. Esto significa que, por defecto, nuestro transistor estará en el estado apagado, y esta resistencia facilitará la labor. Con el fin de comprender mejor el funcionamiento, se muestra en la siguiente figura un ejemplo práctico.



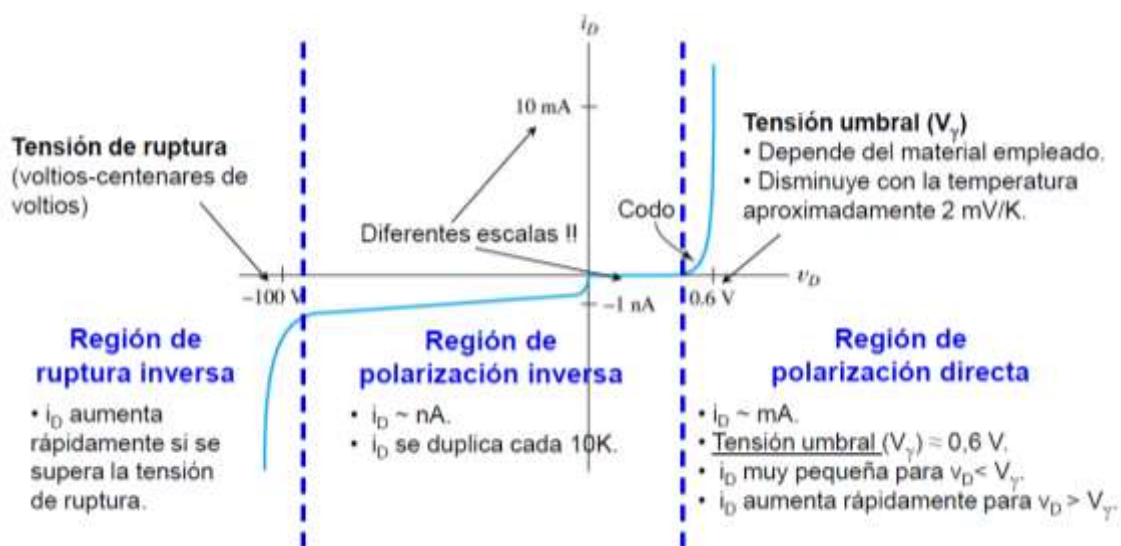
**Figura 23.** Aclaración visual sobre la utilización de resistencias en configuración *pull-down*.

Si el interruptor mostrado en la **Figura 23** está desactivado, conseguiremos que la corriente de la salida de este, idealmente cero, circule por la resistencia dado que no tiene otro lugar por donde “escapar”. Sin embargo, al activar el interruptor, la corriente fluirá a través del recorrido entre la fuente de alimentación y el pin de entrada, puesto que en el camino hacia la toma de tierra se encontrará una resistencia que se opondrá al paso de corriente.

## Diodo

Además de incluir esta resistencia, será necesario incorporar un diodo posicionado electrónicamente en paralelo con el motor a modo de seguridad. Si concebimos de forma electrónicamente pura el motor, veremos que está formado principalmente por un bobinado que será el que produzca una fuerza electromagnética, la cual, interactuando con los imanes situados en el interior de la cavidad del motor permitirá efectuar el giro deseado en este, accionando el motor. Como consecuencia, podemos decir que, a ojos del circuito electrónico, el motor se comportará como una bobina. Las bobinas, debido a sus propiedades, actuarán evitando cambios bruscos de corriente. Si el transistor se apaga rápidamente, cosa que sucederá, dado que accionaremos el motor con un pulso PWM de frecuencia elevada en algunos casos, entonces, la corriente que aún se encontraría circulando a través de la bobina del motor por un tiempo, fluirá a través del diodo y los valores de corriente terminarán disipándose sin derivar hacia otras partes de la circuitería. En el caso en el que decidiésemos no incorporar el diodo, la corriente almacenada en el bobinado no tendría por donde salir, lo que desembocaría en altos valores de tensión en la patilla del drenador que formará parte del circuito de acondicionamiento y como consecuencia, probablemente llegaría a quemar el transistor.

En cuanto a la selección del diodo, tendremos que recalcar en un inicio las principales características de este componente. La **Figura 24**, obtenida de algunas enseñanzas aprendidas durante la carrera [25], trata de mostrar las características teóricas de corriente y voltaje en un diodo, así como sus regiones de funcionamiento.



**Figura 24.** Curva característica I-V de un diodo de Si de pequeña señal a 300°K. [25]

El diodo actuará permitiendo el paso de la corriente cuando entre sus polos se supere la tensión umbral y oponiéndose al paso de esta, idealmente 0A, cuando se encuentre en la región de polarización inversa. Sin embargo, habrá que tener cuidado de que el voltaje en las bornas del componente no supere la región de ruptura inversa, dado que en esta región de funcionamiento el diodo volverá a permitir el paso de corriente.

Deberemos tener en cuenta diversos parámetros a la hora de elegir el componente. Inicialmente, y por las características expuestas en su hoja de datos [26], las cuales se muestran en la **Figura 25**, se ha seleccionado el diodo **1N914**.

**ELECTRICAL CHARACTERISTICS** (Values are at  $T_A = 25^\circ\text{C}$  unless otherwise noted) (Note 2)

Symbol	Parameter		Conditions	Min	Max	Unit
$V_R$	Breakdown Voltage		$I_R = 100 \mu\text{A}$	100		V
			$I_R = 5.0 \mu\text{A}$	75		V
$V_F$	Forward Voltage	914B / 4448	$I_F = 5.0 \text{ mA}$	0.62	0.72	V
		916B	$I_F = 5.0 \text{ mA}$	0.63	0.73	V
		914 / 916 / 4148	$I_F = 10 \text{ mA}$		1.0	V
		914A / 916A	$I_F = 20 \text{ mA}$		1.0	V
		916B	$I_F = 20 \text{ mA}$		1.0	V
		914B / 4448	$I_F = 100 \text{ mA}$		1.0	V
$I_R$	Reverse Leakage		$V_R = 20 \text{ V}$		0.025	$\mu\text{A}$
			$V_R = 20 \text{ V}, T_A = 150^\circ\text{C}$		50	$\mu\text{A}$
			$V_R = 75 \text{ V}$		5.0	$\mu\text{A}$
$C_T$	Total Capacitance	916/916A/916B/4448	$V_R = 0, f = 1.0 \text{ MHz}$		2.0	pF
		914/914A/914B/4148	$V_R = 0, f = 1.0 \text{ MHz}$		4.0	pF
$t_{rr}$	Reverse Recovery Time		$I_F = 10 \text{ mA}, V_R = 6.0 \text{ V (600 mA)}$ $I_R = 1.0 \text{ mA}, R_L = 100 \Omega$		4.0	ns

**Figura 25.** Principales características del diodo 1N914. [26]

Como podemos observar, su tensión de ruptura no será un problema dado que comienza a partir de los 100V y nunca llegaremos a estos valores en nuestra configuración circuital. Además, la corriente en inversa posee unos valores muy bajos que podrán considerarse a efectos prácticos como un diodo ideal. A lo anterior se suma que, su tensión en directa máxima, será de 1V. A partir de estos valores comenzará a permitir el paso de corriente, evitando que la corriente que pueda fluir a través motor, cuando el transistor no esté accionado, consiga disiparse. También debemos de resaltar que el valor de 1V no será suficiente como para quemar el transistor.

Finalmente, en lo referente al diodo, nos centraremos en su tiempo de recuperación en inversa. La idea será accionar el motor solenoide con un pulso PWM, para el modo automático de disparo, que irá directamente al transistor. Por lo que, los accionamientos se producirán mediante cambios bruscos. Si nos fijamos en la hoja de datos del

microcontrolador empleado, se especifican las frecuencias de los pulsos PWM generados, estos pueden apreciarse en la siguiente figura.

The PWM frequency for the output can be calculated by the following equation:

$$f_{OCnxPWM} = \frac{f_{clk\_I/O}}{N \cdot 256}$$

The  $N$  variable represents the prescaler factor (1, 8, 64, 256, or 1024).

**Figura 26.** Frecuencia de oscilación de los pulsos PWM generados en el microcontrolador. [15]

Teniendo en cuenta que la frecuencia de la señal de reloj será de 16MHz, la frecuencia máxima configurable para un pulso PWM será:

$$f_{OCnxPWM} = \frac{f_{clk\_I/O}}{N * 256} = \frac{16MHz}{256} = 62.5KHz$$

Por lo que a esta frecuencia necesitaríamos un tiempo de recuperación de:

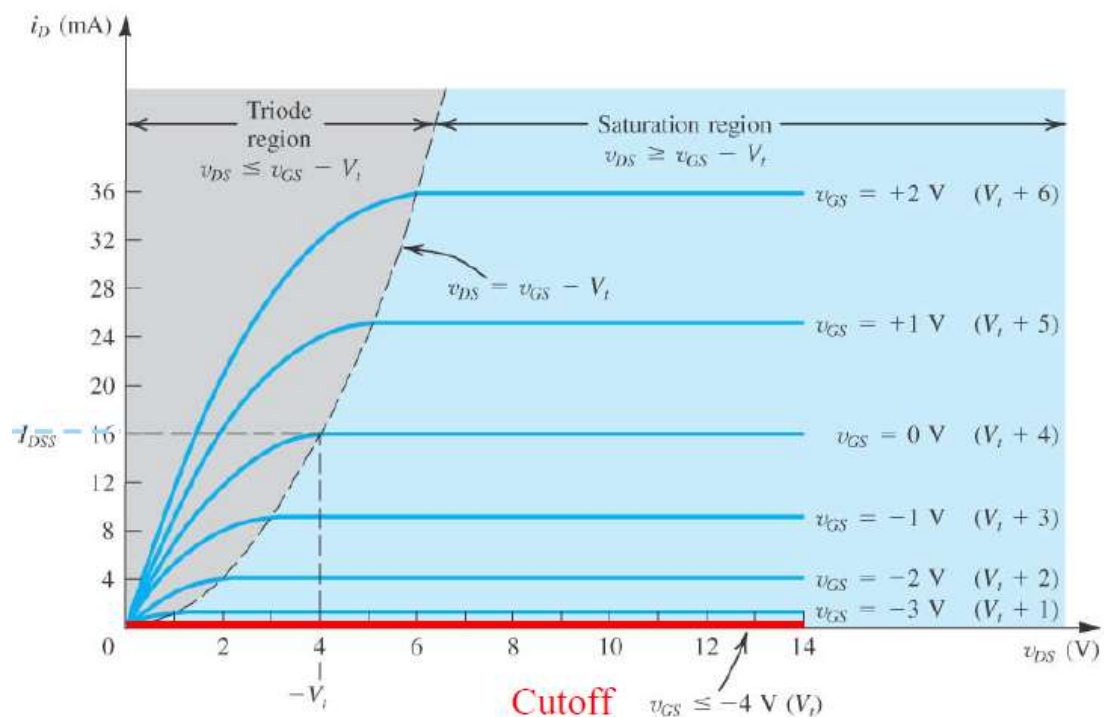
$$T_{RR} = \frac{1}{f_{OCnxPWM}} = \frac{1}{62.5KHz} = 16\mu s$$

Esto sucedería si la señal proviniese directamente del pin PWM asociado y teniendo en cuenta que utilizásemos la frecuencia máxima, cosa que no sucederá. Sin embargo, vemos que el tiempo de recuperación del diodo, tal y como se muestra en la **Figura 26**, es inferior al tiempo de recuperación requerido en el peor de los casos, siendo este de 4ns.

### **Transistor**

Será necesario incorporar un transistor, ya que, esto nos permitirá controlar el motor con el tren de pulsos PWM de una manera eficaz, además, de esta forma podremos incorporar, si fuese necesario en nuevas versiones del producto, fuentes de alimentación externa de altos valores de voltaje. De este modo, la electrónica incorporada nos permitiría accionar los motores con pequeños valores de tensión provenientes de los pines del microcontrolador, pero empleando fuentes de alimentación de valores elevados para los mismos.

A la hora de emplear el transistor debemos recordar algunas de sus características generales con el fin de realizar su implementación como “interruptor”. La **Figura 27**, proveniente de [27], retrata alguna de las propiedades teóricas básicas de un transistor de efecto de campo.



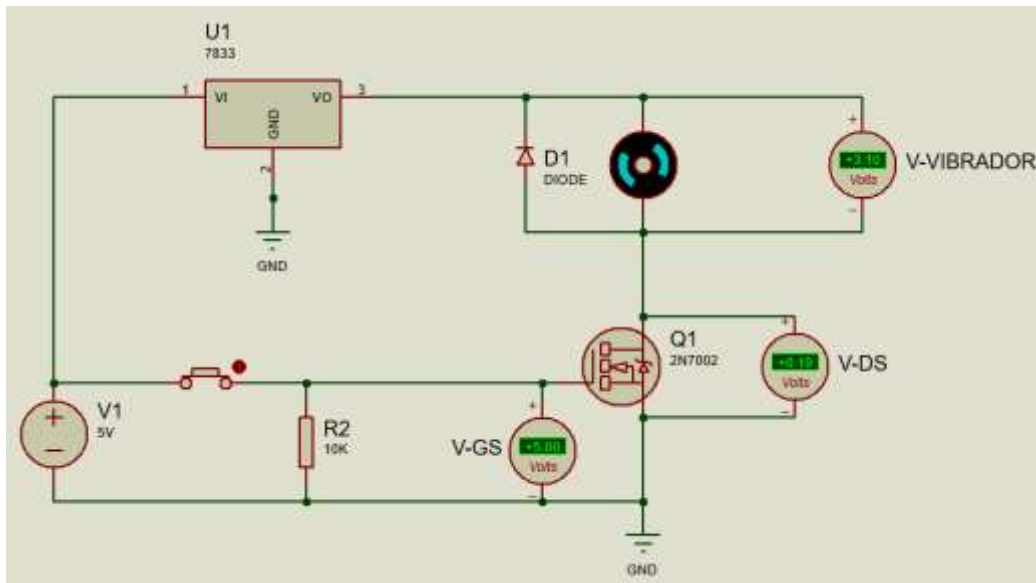
**Figura 27.** Curva característica de un transistor MOSFET de canal N. [27]

Para conseguir que el transistor actúe como un interruptor tendremos que realizar la configuración electrónica de nuestro circuito de forma que esta permita al transistor actuar en las regiones de corte y saturación.

Para situar el transistor en la región de saturación, será necesario que el voltaje drenador fuente sea mayor que el voltaje puerta fuente menos el voltaje umbral en un transistor de canal N. Mientras que para situarnos en la región de corte será necesario que el voltaje entre drenador y fuente sea mayor que el voltaje umbral. Teniendo en cuenta las ecuaciones, deberá cumplirse  $V_{GS} \geq V_{TH}$ ,  $V_{DS} \geq V_{GS} - V_{TH}$  para la región de saturación y  $V_{DS} \geq V_{TH}$  para la de corte.

Con el fin de verificar el funcionamiento del circuito se ha diseñado un pequeño modelo esquemático que ayuda a aclarar la idea, permitiendo ejecutar una simulación. Para este esquemático se ha empleado el transistor lógico de canal N **2N7002** [28]. Dado que algunos de los componentes empleados en los esquemáticos del programa Proteus no poseen sus propios modelos, se emplearán para este pequeño ejemplo, modelos genéricos que permitan ejecutar la simulación del circuito. Continuando con el desarrollo y argumentación del circuito teórico-práctico, se muestra en la **Figura 28** una captura de la simulación del esquemático implementado mediante el programa Proteus.





**Figura 28.** Simulación del circuito de adaptación para conexión con motor realizada con Proteus 8.9

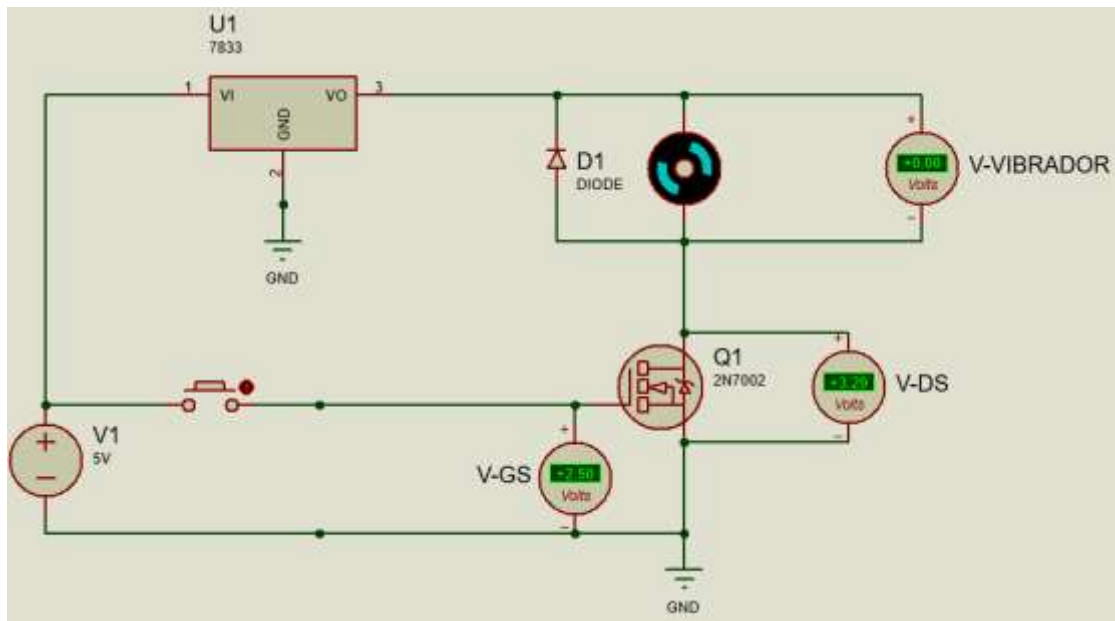
Destacar que, se ha empleado un transistor MOSFET de acumulación de canal N. Se ha tomado esta decisión dado que los MOSFET de acumulación tienen una línea discontinua que conecta fuente y drenador, que indica que el canal no existe para polarización de puerta nula, y por lo tanto será necesario polarizar la puerta para que el canal se forme y permita el paso de los electrones. En este caso emplearemos un MOSFET conocido como “lógico”, este nombre viene dado debido a que puede activarse la circulación de electrones entre drenador y fuente con un voltaje inferior a 5 voltios, o en otras palabras que  $5V \geq V_{TH}$ .

Contando con que la salida del pin PWM de nuestro controlador será de 5V, oscilando entre 0 y 5V (Unipolar), se ha incorporado una fuente de alimentación de 5V, que aportará energía al circuito, y al regulador de voltaje de 3.3V de salida.

También se ha incorporado una resistencia de *pull-down* de 10KOhm que ayudará a mantener la salida del pin PWM del microcontrolador a cero voltios cuando se desactive el MOSFET. En el caso de no incluirse, habría una tensión puerta fuente que podría activar el transistor según se muestra en la **Figura 29**.

No será necesario incluir ningún diodo Zener a mayores con el fin de mejorar la respuesta frecuencial del transistor, dado que este en su propia construcción ya incluye uno entre drenador y fuente.

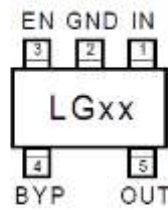
La conexión con el motor solenoide será similar, con la salvedad de eliminar el regulador de voltaje y conectarse directamente a la alimentación del circuito de 5V. A mayores, se sustituirá el transistor **2N7002**, por el modelo **NDS331N** que soporta una tensión de drenador máxima de 1.3A según se muestra en su hoja de datos [29]. Este cambio es de vital importancia, dado que el transistor 2N7002 únicamente cuenta con una corriente de drenador continua máxima de 115mA y no permitiría accionar el motor solenoide, ya que trabaja a 1A. Respecto al resto de características, son similares en el caso de ambos transistores.



**Figura 29.** Representación gráfica que muestra la importancia de la resistencia pull-down mediante simulación en Proteus 8.9

### **Regulador de tensión**

Tratando de llevar a cabo el conexionado del motor vibrador, será necesario emplear un regulador de voltaje, dado que la tensión de alimentación de este motor deberá encontrarse a 3.3V. En este caso, utilizaremos el regulador de voltaje **MIC5219-3.3BM5**, el cual nos permitirá obtener una salida a 3.3V cuando se le aporte una entrada a 5V. En la siguiente figura se incluye un esquema, proveniente de su hoja de datos [30], de las conexiones del componente.



**MIC5219-x.xBM5 / SOT-23-5**  
**Fixed Voltages**  
**(Top View)**

**Figura 30.** Patillaje del regulador de tensión MIC5219-3.3BM5. [30]

En cuanto a sus conexiones, también se aporta una tabla con sus definiciones, proveniente de su *datasheet*. La siguiente figura muestra la tabla anteriormente citada.

### Pin Description

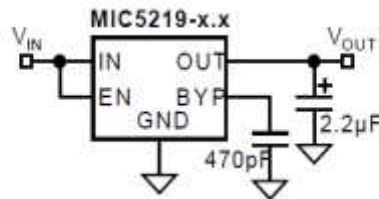
Pin No. MLF-6 TMLF-6	Pin No. MSOP-8	Pin No. SOT-23-5	Pin Name	Pin Function
3	2	1	IN	Supply Input.
2	5–8	2	GND	Ground. MSOP-8 pins 5 through 8 are internally connected.
4	3	5	OUT	Regulator Output.
1	1	3	EN	Enable (Input): CMOS compatible control input. Logic high = enable; logic low or open = shutdown.
6	4 (fixed)	4 (fixed)	BYP	Reference Bypass: Connect external 470pF capacitor to GND to reduce output noise. May be left open.
5(NC)	4 (adj.)	4 (adj.)	ADJ	Adjust (Input): Feedback input. Connect to resistive voltage-divider network.
EP	—	—	GND	Ground: Internally connected to the exposed pad. Connect externally to GND pin.

**Figura 31.** Tabla ilustrativa sobre la funcionalidad de los distintos pines del regulador. [30]

Como podemos observar, tendremos que habilitar la entrada EN para activar el regulador de tensión, además incluiremos un condensador de 470pF con el fin de reducir el ruido de la señal de entrada, proveniente del USB.

En la hoja de datos se aportan algunos ejemplos con una reducción de ruido significativa. Nosotros implementaremos el modelo siguiendo las conexiones mostradas en la **Figura 32**. Aunque según se indica su hoja de datos: "*For applications < 3V, the output capacitor should be increased to 22μF minimum to reduce start-up overshoot. Larger values improve the regulator's transient response. The output capacitor value may be increased without limit.*" [30]. Por este motivo nos decantaremos por un condensador de salida de 22μF mejorando la respuesta en régimen transitorio (velocidad a la que se alcanza un estado estacionario) y salvaguardando el sistema frente a posibles bajadas de tensión, dado que estaremos operando con 3.3V. Además, se recomienda emplear condensadores

electrolíticos de aluminio "The output capacitor should have an ESR (equivalent series resistance) of about  $1\Omega$  or less and a resonant frequency above 1MHz. Ultra-low-ESR capacitors could cause oscillation and/or underdamped transient response. Most tantalum or aluminum electrolytic capacitors are adequate." [30]. La utilización de estos últimos no será empleada, debido al gran tamaño de los condensadores electrolíticos. Además, tal y cómo se indicó en el apartado 3.2.2.1, se añadirá un condensador de 100nF en la señal de entrada proveniente de la alimentación USB.



**Figure 6. Ultra-Low-Noise Fixed Voltage Regulator**

Figure 6 includes the optional 470pF noise bypass capacitor between BYF and GND to reduce output noise. Note that the minimum value of  $C_{OUT}$  must be increased when the bypass capacitor is used.

**Figura 32.** Ejemplo de aplicación del regulador de tensión para condiciones de muy bajo ruido. [30]

### 3.2.2.4 Conexión con diodos LED

En el caso de los diodos LED que simulen el destello lumínico emulando la explosión que se lleva a cabo en el cañón, serán configurados como activos en alta.

Será necesario adaptar la entrada de tensión a los diodos mediante resistencias. Según se indica en la hoja de datos del microcontrolador "each I/O port can sink more than the test conditions (20mA at VCC = 5V, 10mA at VCC = 3V)" [15], siendo el valor máximo de 40mA por pin. Además, el voltaje de un pin digital en alta vendrá determinado por los valores de la siguiente figura, siendo 4.2 el valor mínimo de voltaje para un pin configurado en HIGH.

$V_{OL}$	Output Low Voltage <sup>(3)</sup> ,	$I_{OL} = 10\text{mA}, V_{CC} = 5\text{V}$ $I_{OL} = 5\text{mA}, V_{CC} = 3\text{V}$			0.7 0.5
$V_{OH}$	Output High Voltage <sup>(4)</sup> ,	$I_{OH} = -10\text{mA}, V_{CC} = 5\text{V}$ $I_{OH} = -5\text{mA}, V_{CC} = 3\text{V}$	4.2		2.3

**Figura 33.** Voltaje de salida para datos en HIGH y LOW del microcontrolador Atmega32u4. [15]



### 3.2.2.6 Conexión con interruptores y botones

En lo relativo a las conexiones con los interruptores de final de carrera y botones, estas serán simples, dado que únicamente funcionarán como interruptores y el resto de carga de trabajo correrá por parte del *software*. Bastará con conectar una de las patillas del componente a uno de los pines digitales del microcontrolador y otra a tierra. De esta manera y aplicando la configuración del pin digital en el estado *pull-up*, el pin estará siempre en el estado de la tensión de alimentación y cuando el circuito se cierre porque el interruptor ha sido accionado, el pin que estará en el estado de lectura, detectará la caída de tensión permitiendo actuar al *software* en consecuencia.

### 3.2.2.7 Otras conexiones y conectores

#### *Bootloader*

Con el fin de implementar la carga de la secuencia de arranque o *bootloader* en cada inicio es necesario conectar el pin HWB con una resistencia a tierra. Según se indica en la hoja de datos del microcontrolador “*HWB allows to execute the bootloader section after reset when tied to ground during external reset pulse. The HWB mode of this pin is active only when the HWBE fuse is enable. During normal operation (excluded Reset), this pin acts as a general purpose I/O.*” [15], con lo cual será necesario habilitar este pin para que cada vez que reciba corriente de alimentación cargue el *bootloader*. Esto nos permitirá poder programarlo sin emplear la interfaz ISP cada vez que queramos modificar el código del programa. Al tratarse de una primera versión del producto permitiremos este “fallo de seguridad” ya que, de esta forma, cualquier usuario podrá modificar el código del programa desde la interfaz serie USB. Sin embargo, esta utilización nos dará la posibilidad de programar un reinicio por *software* pudiendo continuar con el funcionamiento típico.

La implementación de este pin a tierra viene determinada porque el pin PE2 HWB es un pin de activa en baja.

#### *ISP*

Teniendo en mente el objetivo de cargar, tras la primera construcción del diseño electrónico, el gestor de arranque, será necesario disponer de una interfaz que nos permita

llevar a cabo esta tarea. Para ello, se ha creado en el diseño esquemático un conector ISP que nos facilitará la instalación del *bootloader*.

Al haber decidido utilizar las facilidades del *software* Arduino, podremos emplear su IDE para realizar la incorporación de la memoria de arranque en nuestro microcontrolador, en [32] se especifica la configuración del puerto *In-Circuit Serial Programming* (ICSP). La **Figura 35** muestra el esquema de conexiones aportado por Arduino para realizar la comunicación ICSP.



**Figura 35.** Esquema de conexiones para ICSP Arduino. [32]

### **Conectores**

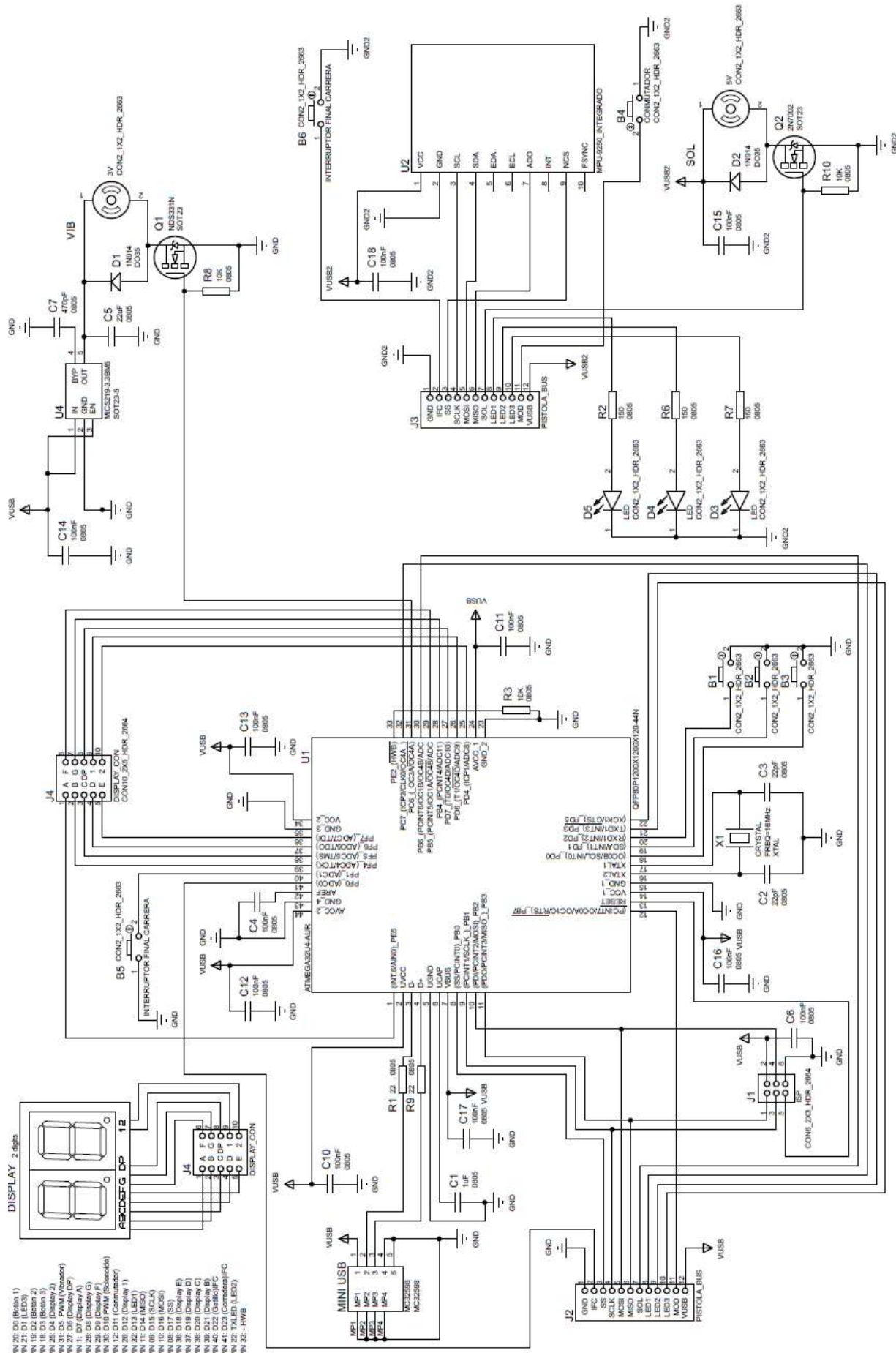
De cara a obtener un diseño compacto y organizado, se ha decidido segmentar el diseño circuital en dos partes que vendrán unidas mediante cableado a través de dos conectores. Este tipo de configuración nos permitirá situar los componentes de forma sencilla en el lugar que les corresponda, pudiendo así situar, el motor solenoide, los diodos LED, el interruptor de la corredera y los sensores en la carcasa del arma correspondiente a la sección del cañón.

### **3.2.2.8 Implementación final**

Como punto final de la creación del desarrollo electrónico esquemático se aporta el resultado final desarrollado con el programa Proteus 8.9, donde pueden visualizarse las conexiones de manera detallada, así como las huellas de los componentes (aportados sus datos junto a los distintos elementos) y los valores de los distintos componentes electrónicos. Se aporta a mayores un resumen en el que figuran los números de los pines empleados junto a su función.

Con la finalidad de relajar la dificultad del desarrollo *software*, se incluye además, en este resumen, la correspondencia en pines con la biblioteca de Arduino IDE correspondiente con el integrado Arduino Micro, el cual cuenta con el mismo microcontrolador que hemos empleado durante el desarrollo electrónico.

- PN 25: D0 (Botón 1)
- PN 27: D1 (LED3)
- PN 19: D2 (Botón 2)
- PN 18: D3 (Botón 3)
- PN 25: D4 (Display 3)
- PN 27: D5 (Display 4)
- PN 1: D7 (Display A)
- PN 26: D8 (Display G)
- PN 29: D9 (Display F)
- PN 28: D10 (Display E)
- PN 12: D11 (Comunicador)
- PN 26: D12 (Display 1)
- PN 22: D13 (LED1)
- PN 23: D14 (LED2)
- PN 10: D15 (MOSI)
- PN 08: D16 (SS)
- PN 08: D17 (SS)
- PN 36: D18 (Display E)
- PN 36: D19 (Display C)
- PN 36: D20 (Display B)
- PN 36: D21 (Display A)
- PN 40: D22 (Gatillo/PC)
- PN 41: D23 (Comunicador)
- PN 42: D24 (LED4)
- PN 33: HWB





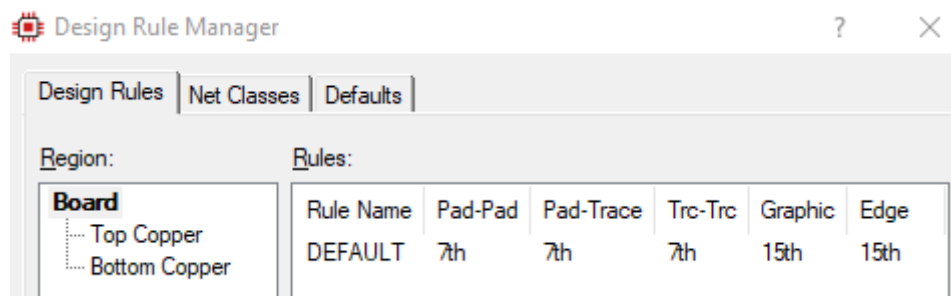
### 3.3 Diseño de la PCB

Una vez definido el diseño esquemático, con las respectivas huellas de los distintos componentes, se ha procedido a realizar el diseño de la PCB. Para ello se ha continuado utilizando el programa Proteus 8.9.

Con el objetivo de reducir al máximo la superficie de la PCB se ha elaborado el rutado de conexiones en un marco de trabajo de 5 x 5cm.

Inicialmente se ha dado pie al establecimiento de los criterios de diseño, con el fin de evitar posibles desajustes en las medidas, para ello se han establecido las reglas de diseño mostradas en la *Figura 36*.

Tras esto, se ha procedido a incorporar los agujeros de la placa, los cuales servirán para ajustar parte de la PCB en la carcasa plástica. Se han definido de un tamaño de 3mm de



*Figura 36. Reglas de diseño fijadas durante la elaboración del diseño de la PCB*

diámetro con la finalidad de poder pasar a través de los mismos tornillería de tipo M3.

Posteriormente se han generado planos de tierra en la capa inferior mediante la herramienta *power plane generator* con el objetivo de mitigar las *ElectroMagnetic Interference* (EMI) y evitar bucles de puesta a tierra que provoquen que las “descargas” a tierra no se realicen de manera uniforme.

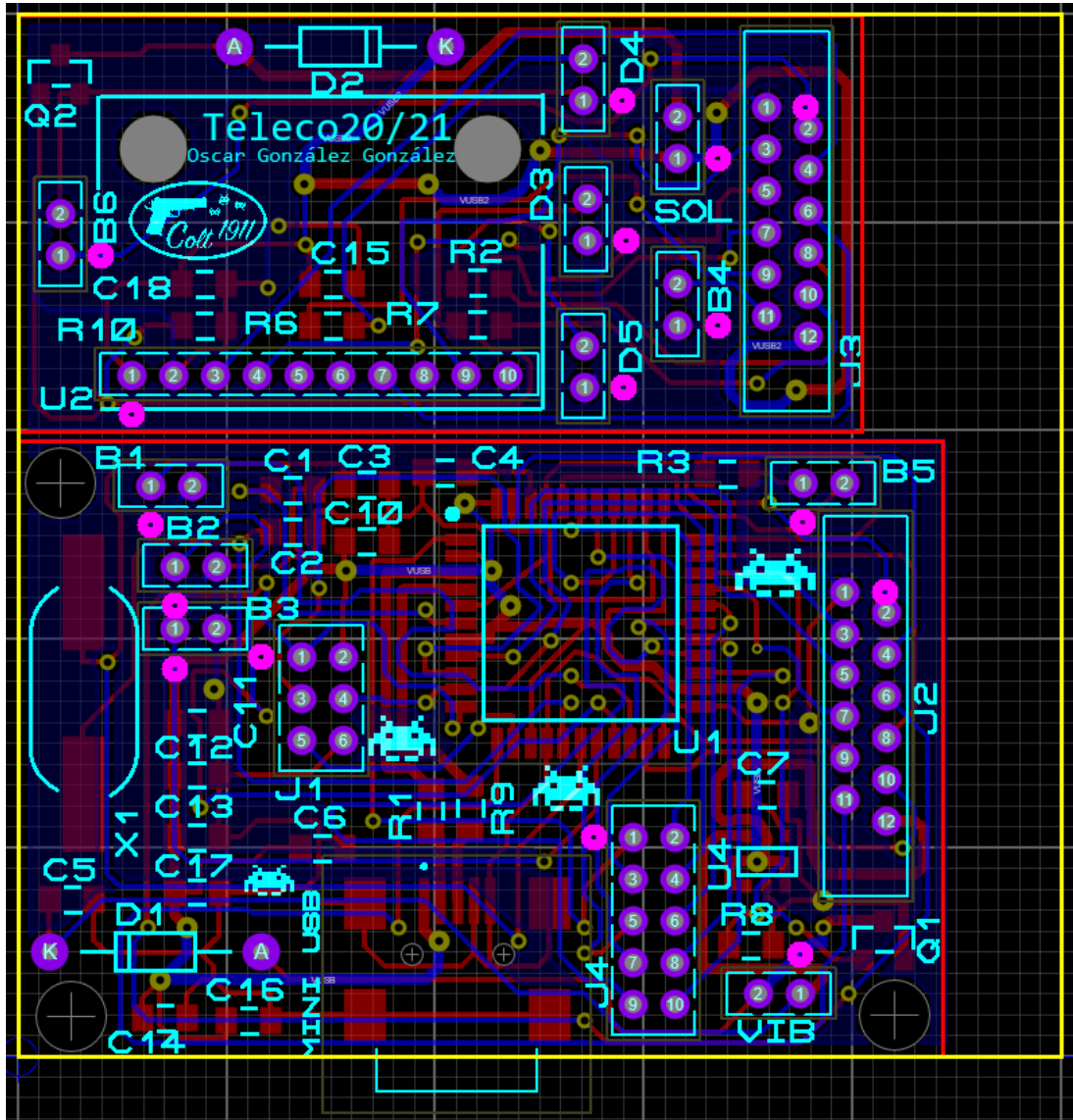
Tras haber definido los márgenes e incorporado los componentes en función del espacio disponible (después de una primera iteración con el diseño de la carcasa que se expondrá en la sección 5), se ha procedido a emplear la herramienta *auto-router* del programa. Una vez realizada la iteración con esta herramienta, debido a las pequeñas dimensiones requeridas para la PCB no se han podido establecer todas las conexiones y algunas de ellas han tenido que implementarse de forma manual.

Después de haber incorporado todas el rutado de los *path*, se ha procedido a engrosar aquellas pistas que fueran a ser cursadas por altos valores de intensidad, a modo de seguridad. Para ello, se han engrosado estas pistas a un valor de T20 y se ha aumentado

el tamaño de las vías a un valor V40. Las rutas en las que se han implementado estas modificaciones han sido las pistas VUSB y GND que se dirigían hacia los motores, dado que serán estos los elementos que mayor potencia consuman en todo el circuito.

Llevadas a la práctica estas modificaciones, se ha procedido a corregir los errores de diseño relativos al incumplimiento de distancias entre elementos como consecuencia del engrosamiento de las pistas y vías.

Finalmente se han incorporado una serigrafía personalizada en la cara superior y se han



*Figura 37. Implementación final del diseño de la PCB*

ajustado las serigrafías de los distintos componentes con la finalidad de que fueran visibles, evitando así huecos de vías, orificios de la PCB y otros elementos. El resultado final tras el diseño de la PCB puede visualizarse en la *Figura 37*.

### 3.4 Análisis de potencia

Con el fin de tratar de establecer una fuente de alimentación que satisfaga todas nuestras necesidades, así como comprender si la composición de nuestro circuito es viable desde el punto de vista del consumo de potencia. Se realizará un análisis de potencia sobre los elementos activos del circuito.

Nuestro producto ha sido concebido para ser alimentado a través de un conector mini USB, con lo cual, el conjunto de nuestro circuito vendrá alimentado a través de conectores USB en principio procedentes de un equipo que generalmente será un ordenador. A lo largo de este documento y según lo expuesto en [16] y [33], trataremos de determinar cuál será una alimentación suficiente para el conjunto de componentes de nuestro integrado.

#### 3.4.1 Conjunto de sensores MEMS

En la PCB integrada que incorpora el sensor MPU-9250 se encuentra un regulador de tensión incorporado que nos permitirá conectar directamente nuestra fuente de alimentación USB a la PCB del MPU-9250, a pesar de que su voltaje de funcionamiento deberá encontrarse entre 2.4-3.6V según se refleja en su hoja de datos. Consultando las características eléctricas del mismo, encontramos la siguiente tabla mostrada en la **Figura 38**, en la que se visualizan los valores requeridos por los sensores MEMS en condiciones de una alimentación a 2.5V y una temperatura de 25°C.

PARAMETER	CONDITIONS	MIN	TYP	MAX	Units	Notes
<b>SUPPLY VOLTAGES</b>						
VDD		2.4	2.5	3.6	V	
VDDIO		1.71	1.8	VDD	V	
<b>SUPPLY CURRENTS</b>						
Normal Mode	9-axis (no DMP), 1 kHz gyro ODR, 4 kHz accel ODR, 8 Hz mag. repetition rate		3.7		mA	
	6-axis (accel + gyro, no DMP), 1 kHz gyro ODR, 4 kHz accel ODR		3.4		mA	
	3-axis Gyroscope only (no DMP), 1 kHz ODR		3.2		mA	
	6-axis (accel + magnetometer, no DMP), 4 kHz accel ODR, mag. repetition rate = 8 Hz		730		µA	
	3-Axis Accelerometer, 4kHz ODR (no DMP)		450		µA	
	3-axis Magnetometer only (no DMP), 8 Hz repetition rate		280		µA	
Accelerometer Low Power Mode (DMP, Gyroscope, Magnetometer disabled)	0.98 Hz update rate		8.4		µA	1
	31.25 Hz update rate		19.8		µA	1
Full Chip Idle Mode Supply Current			8		µA	
<b>TEMPERATURE RANGE</b>						
Specified Temperature Range	Performance parameters are not applicable beyond Specified Temperature Range	-40		+85	°C	

**Figura 38.** Características eléctricas del conjunto de sensores MEMS MPU-9250.[13]

Dado que el vendedor del conjunto integrado no incorpora características de consumo de potencia sobre el regulador de voltaje incorporado, únicamente tendremos en cuenta el consumo de potencia del integrado MPU-9250.

Como podemos observar, en el modo de funcionamiento con los tres sensores, sin emplear el *Digital Motion Processor* (DMP) es de 3.7mA. En el caso de no emplear el magnetómetro y continuar sin utilizar el DPM el consumo será de 3.4mA.

En la práctica, suponiendo que el conversor de voltaje incorporado aporta una salida de 2.5V al integrado MPU-9250, determinaremos que el consumo de potencia de este elemento será de  $5V \cdot 3.7mA = 18,5mW$ . Aunque la utilización del magnetómetro aún no ha sido fijada, nos trataremos de situar en las condiciones más extremas. En principio no emplearemos el DMP dado que nuestro microcontrolador debería de ser capaz de gestionar la información sin mayores problemas.

### 3.4.2 Motor vibrador

En las especificaciones aportadas por el vendedor del motor, se nos muestra como los valores recomendados de funcionamiento son de 3V – 180mA (540mW). Dado que la alimentación del circuito estará fijada a 5V será necesario comprobar si el regulador de tensión es capaz de aportarnos esos valores.

#### 3.4.2.1 Regulador de tensión

En la hoja de datos de nuestro regulador de tensión se indican cómo, para conseguir obtener una corriente de 500mA de salida, será necesaria una caída de voltaje mínima ( $V_{IN} - V_{OUT}$ ) de 350mV. Nosotros cumpliremos de sobra esos requisitos, ya que la caída de tensión entre la entrada y la salida del regulador será de  $5V - 3.3V = 1.7V$  superando así los límites fijados y permitiéndonos alcanzar los 180mA demandados por el motor vibrador.

La potencia recomendada por el vendedor del motor vibrador será de  $3V \cdot 180mA = 540mW$ . Sin embargo, tendremos que tener en cuenta las consideraciones térmicas a la hora de llevar a la práctica estos valores. Según se nos indica en la hoja de datos del regulador **MIC5219-3.3BM5** perteneciente a la serie de productos MIC5219 del fabricante, será necesario contar con las consideraciones térmicas del componente. En su

hoja de datos se muestra la expresión matemática que nos permitirá calcular la potencia disipada en el regulador de tensión.

$$P_D = (V_{IN} - V_{OUT})I_{OUT} + V_{IN}I_{GND} = (5V - 3.3V)180mA + 5V3mA = 321mW$$

Se ha tomado como valor de  $I_{GND}$  3mA debido a que para una tensión de habilitación del regulador superior a 3V (la nuestra será de 5V) y una corriente de salida de 150mA los valores máximos de  $I_{GND}$  serán de 3mA, con lo cual podremos tantear que para 180mA de corriente de salida se aproximará al valor máximo experimentado para 150mA. En el peor de los casos, tomando una corriente de salida de 500mA el valor de  $P_D = 366mW$ . En la hoja de datos de este componente también se refleja cómo realizar el cálculo para la disipación de potencia máxima del elemento.

$$P_D(max) = \frac{(T_J(max) - T_A)}{\theta_{JA}} = \frac{(125^\circ C - 25^\circ C)}{220^\circ C/W} \approx 455mW$$

Donde  $T_J(max)$  es la temperatura máxima de la unión antes de que el componente se quemara,  $T_A$  es la temperatura ambiente establecida en  $25^\circ C$  y  $\theta_{JA}$  es la disipación recomendada en función de la huella del componente, para nuestro ejemplo, al poseer una huella SOT-23-5 será de  $220^\circ C/W$ . Por lo tanto, no superaremos la potencia máxima disipable y podremos emplear el componente con seguridad.

Hemos de indicar que, a pesar de que en las características eléctricas del dispositivo se especifica que los valores numéricos están diseñados para un  $V_{IN} = V_{OUT} + 1V$  nosotros tenemos una  $V_{IN} = V_{OUT} + 1.7V$ , sin embargo, situándonos en las peores condiciones, donde  $I_{OUT} = 500mA$  superaríamos la potencia máxima, pero este efecto no será notable, dado que el motor vibrador no se activará todo el rato.

Finalmente hay que aclarar que, a pesar de que el motor debería trabajar a 3V y estamos empleando un regulador de tensión con un voltaje de salida de 3.3V la implementación circuital del transistor acabará originando que sobre el motor haya una caída de  $\approx 3V$  según se mostró en la simulación del circuito de acondicionamiento del motor.

### 3.4.2.2 Transistor 2N7002

Tras haber comprobado el regulador de tensión, será necesario verificar si el transistor será capaz de disipar la potencia entregada al motor, para ello se volverán a realizar los cálculos anteriores modificando los valores. En la hoja de datos del transistor 2N7002 se

muestra que la resistencia térmica es de  $625^{\circ}\text{K}/\text{W}=625-273.15=351.85^{\circ}\text{C}/\text{W}$  y su temperatura máxima de disipación es de  $150^{\circ}\text{C}$ , con lo cual obtenemos:

$$P_D(\text{max}) = \frac{(T_J(\text{max}) - T_A)}{\theta_{JA}} = \frac{(150^{\circ}\text{C} - 25^{\circ}\text{C})}{351.85^{\circ}\text{C}/\text{W}} \approx 355\text{mW}$$

En principio podríamos pensar que se supera la potencia máxima de disipación, sin embargo, durante la simulación mostrada en la **Figura 23** veíamos cómo únicamente caían  $0.19\text{V}$  en el transistor, con lo cual, suponiendo que a través de este elemento circulan los  $180\text{mA}$  del motor vibrador únicamente recaerá sobre este una potencia de  $0.19\text{V} * 180\text{mA} = 34.2\text{mW}$ , a pesar de esto, la corriente máxima de drenador según su hoja de datos será de  $115\text{mA}$ .

### 3.4.3 Motor solenoide

En la información proporcionada por el vendedor del componente se especifica que sus características I-V deberán ser de  $6\text{V}-1\text{A}$ . Nos encontramos ante valores elevados para este componente, a pesar de que nuestro circuito trabajará a  $5\text{V}$  no ha sido posible encontrar ninguna variante a  $5\text{V}$  que posea un recorrido de  $1\text{cm}$ , por lo que trataremos de emplear este componente, aunque no sea el más adecuado para la situación. La carencia de ese voltaje originará posiblemente una pérdida de fuerza en el movimiento del solenoide, sin embargo, esto no será un problema dado que la pieza plástica de la corredera tendrá un peso ínfimo.

El consumo de potencia de este componente se verá reflejado en la práctica según  $5\text{V} * 1\text{A} = 5\text{W}$ , será pues, el componente que tenga un mayor consumo eléctrico en nuestra configuración electrónica, aunque será necesario realizar pruebas físicas para determinar cómo varía el componente al establecer los valores recomendados en otros inferiores.

Tendremos que verificar si el transistor empleado en el circuito de adaptación es capaz de disipar toda la potencia entregada.

#### 3.4.2.2 Transistor NDS331N

En su hoja de datos se muestra cómo su resistencia térmica a temperatura ambiente es de  $250^{\circ}\text{C}/\text{W}$  y su valor máximo de temperatura es de  $150^{\circ}\text{C}$  con lo cual obtenemos:

$$P_D(max) = \frac{(T_J(max) - T_A)}{\theta_{JA}} = \frac{(150^{\circ}C - 25^{\circ}C)}{250^{\circ}C/W} = 500mW$$

Este valor supera los 5W, sin embargo, al igual que en el caso del transistor del motor solenoide, sobre este elemento únicamente caerá una pequeña parte de los 5W provenientes de la alimentación USB, esto, sumado a que el motor solenoide no será actuado todo el rato será suficiente como para disipar la potencia que caiga sobre este elemento. En el caso del modo de funcionamiento automático la potencia media será incluso menor, dado que el solenoide se actuará con un tren de pulsos.

### 3.4.4 Microcontrolador

Nuestro microcontrolador será el encargado de distribuir la potencia por el resto del circuito, a diversos componentes, es por esto que será necesario comprender cuánta potencia podrá aportar a los distintos elementos del circuito.

A la hora de cuantificar el consumo del microcontrolador nos centraremos inicialmente en el consumo de procesamiento de este. En su hoja de datos se muestra la gráfica aportada en la **Figura 39** en la que se observa la relación entre el consumo de potencia y la frecuencia en condiciones de temperatura ambiente.

Figure 30-4. Active Supply Current vs. Frequency (1 - 16MHz) and T = 25°C

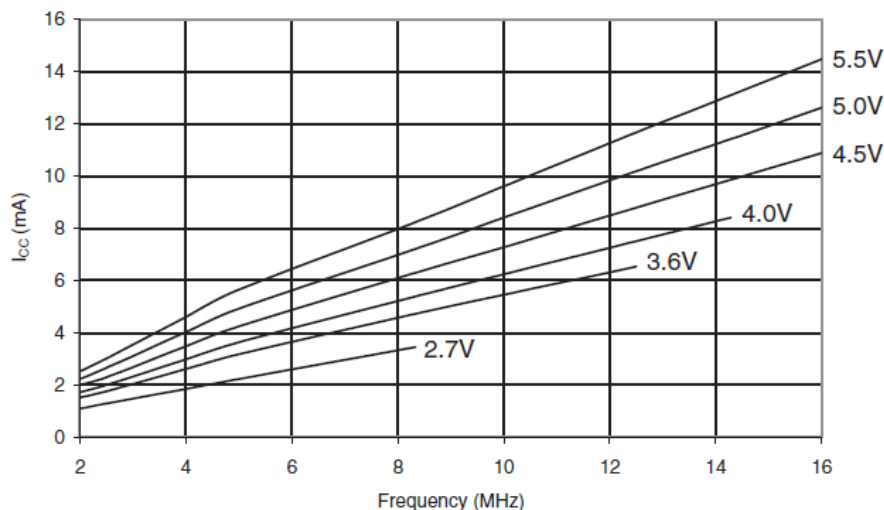


Figura 38. Relación entre consumo de potencia y frecuencia de trabajo ATmega32u4. [15]

Tal y como podemos observar, para una frecuencia de 16Mhz trabajando a 5V tendremos un consumo de corriente de 13mA aproximadamente, con lo que su potencia máxima, sin contar los periféricos será de  $5V * 13mA = 65mW$ .

En cuanto a los periféricos trataremos de analizar los distintos elementos que emplearán las salidas del microcontrolador.

#### **3.4.4.1 Salidas de datos**

Tal y cómo se refleja en su hoja de datos, se ha mostrado con anterioridad, en la *Figura 33* de la sección 3.2.2.4 cuáles serán los valores de las salidas de datos del microcontrolador.

Como podemos observar, las salidas de datos configuradas en alta tendrán un valor máximo de 4.2V y 10mA, mientras que las salidas configuradas en baja tendrán un valor de 0.7V y 10mA.

Es difícil cuantificar cuál será el consumo de los pines de datos, dado que no estarán enviando información todo el tiempo, además de enviar datos en alta y baja. Con lo cual supondremos para el análisis de potencia que poseen una impedancia de salida  $\approx 0$  y una impedancia de entrada  $\approx \infty$ , es por esto que su consumo no será tenido en cuenta, dado que únicamente consumirán potencia durante las transiciones.

#### **3.4.4.2 Entradas de datos de interruptores y botones**

En lo referente a las entradas de datos provenientes de los interruptores y botones, no generarán consumo de potencia significativo, dado que están conectadas a la toma de tierra y serán configuradas en el estado *pull-up*.

#### **3.4.4.3 Salidas hacia diodos**

En cuanto a las salidas de datos destinadas a diodos, estos serán configurados como activos en alta en el caso de los diodos de la corredera y activos en baja para el *display* de dos dígitos de ánodo común.

En lo referente al *display* LED no podremos realizar cálculos de potencia, dado que el vendedor da un rango de tensión de alimentación entre 5V y 24V sin especificar la corriente necesaria.

En cuanto a los LED incorporados en la corredera el vendedor detalla como para una máxima luminosidad deberemos emplear un voltaje de trabajo entre los 3.2 y los 3.4



voltios. Si tenemos en cuenta que la corriente máxima aportada por los pines es de 10mA, obtendremos una potencia de 33mW por cada LED, siendo en total **99mW**.

#### 3.4.4.4 Salidas PWM para los motores

En cuanto a los pulsos PWM que permitirán habilitar los transistores MOSFET tipo N activando así los motores, emplearemos una métrica similar a los pines de salidas de datos realizando la media entre el voltaje en alta y en baja. Si suponemos que empleamos un ciclo de trabajo del 50% tendremos que la media del voltaje empleado será de  $(4.2+0.7)/2=2.45V$  suponiendo que la salida de corriente continúe siendo de 10mA tendremos un consumo de  $2.45V*10mA*2pines=49mW$ . Al ser inferior a la salida de los pines en alta para activación de los motores de forma continua, se despreciará esta potencia en el cómputo total.

#### 3.4.4.5 Otras consideraciones

Es cierto que en nuestro estudio de potencia no se están incluyendo de forma íntegra todos los consumos, sin embargo, se obviarán el resto de elementos no citados durante este análisis, dado que se han realizado los cálculos máximos en situaciones de estrés y frente a estos valores podrían despreciarse otras consideraciones menores.

### 3.4.5 Cómputo global y conclusiones

Con el fin de realizar un estudio aproximado del consumo total se aportará en la siguiente tabla un resumen sobre el consumo de los componentes.

Sensores	9.25mW
Motor vibrador	540mW
Motor solenoide	5W
Microcontrolador	65mW
Periféricos microcontrolador	99mW (LED)
Potencia total necesaria	5.71W
Amperaje total necesario	$5.71W/5V=1.142A$

Según se cita en [33] “*The VBUS Current Handling Capability field (B6...5) Shall indicate whether the cable is capable of carrying default current (500mA USB2, 900mA USB3.2 x1, 1.5A USB3.2 x2)*”, por tanto, como conclusiones a los resultados obtenidos

queda descartada la idea de emplear un único conector USB 2.0 el cual posee una corriente máxima de 500mA [1]. El empleo de un único conector USB 3.0-3.2 para la toma de corriente tampoco sería una solución completamente eficiente siendo su corriente máxima 900mA [2].

Finalmente se ha establecido la propuesta de emplear un cable USB 3.2 tipo A – mini USB tipo B. De esta manera se podrían obtener 900mA en caso de conectarse a un puerto USB 3.2. A mayores, el producto deberá incluir un adaptador como el mostrado en la **Figura 39**.



**Figura 39.** Adaptador dual USB 3.0 datos y carga - USB 2.0 carga

De esta forma se podría utilizar el producto con un solo cable USB 3.0 en caso de no contar con varios puertos, obteniendo una corriente máxima de 900mA. También se podría emplear el adaptador en dos USB 2.0 obteniendo una corriente máxima de 1A, en caso de no contar con puertos USB 3.0. En estos dos casos se vería reducida la fuerza de los motores.

Finalmente se podría emplear una última configuración que permitiera emplear el producto a pleno rendimiento empleando el adaptador en un USB 3.0 y otro USB 2.0 consiguiendo una corriente máxima de 1.4A.

En la práctica, tras haber realizado mediciones y comprobaciones físicas, será posible emplear únicamente un conector USB 2.0 y como contrapartida obtendremos una menor

fuerza en los motores. Esto sucede porque los cálculos realizados están pensados de cara a una situación crítica en la que se emplee toda la potencia posible de manera simultánea y esto prácticamente no sucederá durante su utilización.

## 3.5 Obtención de materiales

Durante el transcurso de esta sección se aclarará el desarrollo de la *Bill Of Materials* (BOM), así como otros aspectos relacionados con el proceso de obtención de los materiales incorporados en la construcción de nuestro producto.

### 3.5.1 BOM

A la hora de definir la factura de los materiales se ha elaborado una tabla en la que figuran distintos aspectos económicos de interés como el precio de las unidades individuales, la inversión total de cara a la obtención de los componentes electrónicos, pero también los costes originados tras la adquisición de elementos complementarios inmiscuidos en el desarrollo del producto.

El listado de materiales completo generado con el programa Proteus 8.9 puede visualizarse en las siguientes páginas de este documento. El documento BOM generado mediante Proteus únicamente incluye el precio por componente (sin gastos de envío), y no la inversión total necesaria, dado que los componentes se suelen vender de forma habitual en paquetes de un gran número de unidades. A continuación, se expone en la **Figura 40** una tabla explicativa que incluye todos los elementos necesarios, incluyendo la PCB, los cables y otros componentes que han sido precisos para el desarrollo del producto.

COMPONENTE	MODELO	CANTIDAD	PRECIO POR UNIDAD	GASTOS DE ENVÍO	PRECIO TOTAL
Microcontrolador	ATmega32U4-AU	5	4,484	0	22,42
Conjunto integrado de sensores MEMS	MPU-9250	1	3,66	1,34	5
Transistor de efecto de campo	2N7002	100	0,0067	0,81	1,48
Transistor de efecto de campo	NDS331N	200	0,03185	1,61	7,98
Regulador de tensión	MIC5219-3.3BM5	10	0,057	0,54	1,11
Conector de mini USB 5 pines	Mini USB tipo B	10	0,072	0,48	1,2
Micro-motor vibrador DC	N/A	5	1,018	0,63	5,72
Motor Solenoide	HS-0730B	1	7,79	0	7,79
Diodo	1N914	50	0,0064	0,55	0,87
Oscilador de cristal 16MHz	N/A	10	0,062	0,54	1,16
Interruptor de final de carrera	T403	35	0,342571429	0	11,99
Condensador cerámico	22uF	30	0,061333333	1,52	3,36
Condensador cerámico	1uF	100	0,0184	1,52	3,36
Condensador cerámico	100nF - 0.1uF	100	0,0184	1,52	3,36
Condensador cerámico	22pF	100	0,0184	1,52	3,36
Condensador cerámico	470pF	100	0,0184	1,52	3,36
Resistencias	220hm	100	0,004	0,7	1,1
Resistencias	10kOhm	100	0,004	0,7	1,1
LED	Orange	50	0,0226	0,46	1,59
Resistencias	150Ohm	100	0,004	0,7	1,1
Botones	Color Negro	10	0,044	0,48	0,92
Display LED 2 dígitos	Ánodo común 2 dígitos	20	0,399	0	7,98
Conmutador	6MM	20	0,025	0,48	0,98
PCB	N/A	5	1	25	30
Cable USB 3.0 a USB mini tipo B	N/A	1	1,44	0	1,44
Adaptador de corriente dual USB	N/A	1	2,35	0,53	2,88
Cable USB 3.0 macho a macho	N/A	1	4,67	0	4,67
Programador USBASP-AVR	N/A	1	3,66	1	4,66
Multímetro USB	N/A	1	7,88	3,06	10,94
				<b>INVERSION INICIAL</b>	<b>152,88</b>

Figura 40. Tabla resumen con costes generales e inversión inicial

## Bill Of Materials for Gun Controller

**Design Title** Gun Controller  
**Author** Oscar González González  
**Document Number** 1  
**Revision** 1  
**Design Created** domingo, 28 de marzo de 2021  
**Design Last Modified** sábado, 10 de julio de 2021  
**Total Parts In Design** 49

### 16 Capacitors

Quantity	References	Value	PCB Package	Stock Code	Unit Cost
1	C1	1uF	0805	<a href="#">Aliexpress/item=1005001950625934</a>	€0,01840
2	C2-C3	22pF	0805	<a href="#">Aliexpress/item=1005001950625934</a>	€0,01840
11	C4,C6,C10-C18	100nF	0805	<a href="#">Aliexpress/item=1005001950625934</a>	€0,01840
1	C5	22uF	0805	<a href="#">Aliexpress/item=1005001950625934</a>	€0,06133
1	C7	470pF	0805	<a href="#">Aliexpress/item=1005001950625934</a>	€0,01840

Sub-totals: €0,33733

### 8 Resistors

Quantity	References	Value	PCB Package	Stock Code	Unit Cost
2	R1,R9	22	0805	<a href="#">Aliexpress/item=32858225842</a>	€0,00400
3	R2,R6-R7	150	0805	<a href="#">Aliexpress/item=32858225842</a>	€0,00400
3	R3,R8,R10	10K	0805	<a href="#">Aliexpress/item=32858225842</a>	€0,00400

Sub-totals: €0,03200

### 3 Integrated Circuits

Quantity	References	Value	PCB Package	Stock Code	Unit Cost
1	U1	ATMEGA32U4-AUR	QFP80P1200X1200X120-44N	<a href="#">Aliexpress/item=32916694199</a>	€4,48400
1	U2	MPU-9250_INTEGRADO	CON10_1X10_HDR_2663	<a href="#">Aliexpress/item=4000093478742</a>	€3,66000
1	U4	MIC5219-3.3BM5	SOT23-5	<a href="#">Aliexpress/item=1005001810964057</a>	€0,05700

Sub-totals: €8,20100

### 2 Transistors

Quantity	References	Value	PCB Package	Stock Code	Unit Cost
1	Q1	NDS331N	SOT23	<a href="#">Aliexpress/item=4000161823022</a>	€0,03185
1	Q2	2N7002	SOT23	<a href="#">Aliexpress/item=32843479156</a>	€0,00670

Sub-totals: €0,03855

### 5 Diodes

Quantity	References	Value	PCB Package	Stock Code	Unit Cost
2	D1-D2	1N914	DO35	<a href="#">Aliexpress/item=1005002192687278</a>	€0,00640
3	D3-D5	LED	CON2_1X2_HDR_2663	<a href="#">Aliexpress/item=4000896062123</a>	€0,02260

Sub-totals: €0,08060

### 15 Miscellaneous

Quantity	References	Value	PCB Package	Stock Code	Unit Cost
3	B1-B3	BOTÓN	CON2_1X2_HDR_2663	<a href="#">Aliexpress/item=4001068725725</a>	€0,04400
1	B4	CONMUTADOR	CON2_1X2_HDR_2663	<a href="#">Aliexpress/item=4001207529493</a>	€0,02500
2	B5-B6	INTERRUPTOR FINAL CARRERA	CON2_1X2_HDR_2663	<a href="#">Amazon/ASIN=B08LD68FYQ</a>	€0,34257
1	DISPLAY	2 digits	NULL	<a href="#">Aliexpress/item=4000939769059</a>	€0,39900
1	J1	ISP	CON6_2X3_HDR_2664	N/A	€0,00000
2	J2-J3	PISTOLA_BUS	CON12_1X12_UB_58JF	N/A	€0,00000
1	J4	DISPLAY_CON	CON10_2X5_HDR_2664	N/A	€0,00000

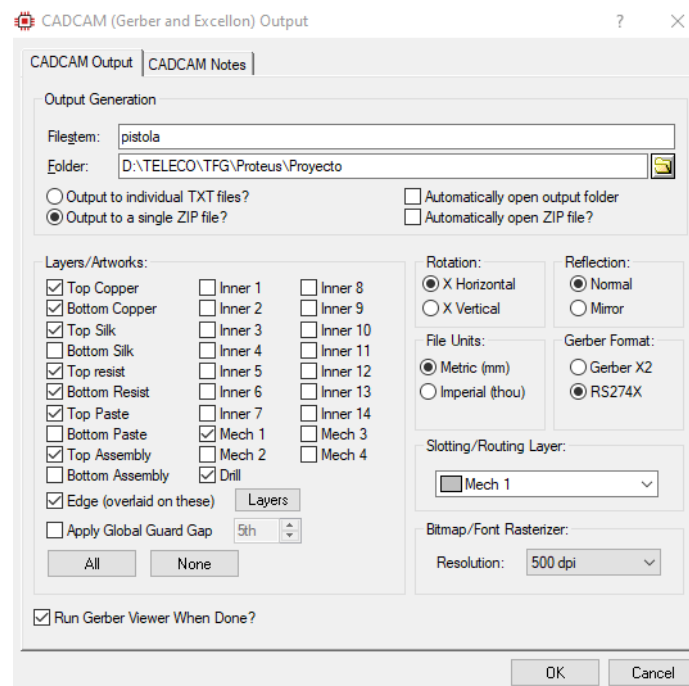
1	MINI USB	MC32598	MC32598	<a href="https://www.aliexpress.com/item/32696841451">Aliexpress/item=32696841451</a>	€0,07200
1	SOL	5V	CON2_1X2_HDR_2663	<a href="https://www.amazon.com/dp/B07MCFB772">Amazon/ASIN=B07MCFB772</a>	€7,79000
1	VIB	3V	CON2_1X2_HDR_2663	<a href="https://www.aliexpress.com/item/1005001297087992">Aliexpress/item=1005001297087992</a>	€1,01800
1	X1	CRYSTAL	XTAL	<a href="https://www.aliexpress.com/item/32923963331">Aliexpress/item=32923963331</a>	€0,06200
<b>Sub-totals:</b>					<b>€10,18314</b>
<b>Totals:</b>					<b>€18,87263</b>

lunes, 19 de julio de 2021 18:03:05

En la sección 1 del anexo que incorpora este documento se incluirán algunos detalles sobre los anuncios de compra y enlaces, con el fin de no sobrecargar la parte principal de la memoria.

### 3.5.2 Adquisición de PCB

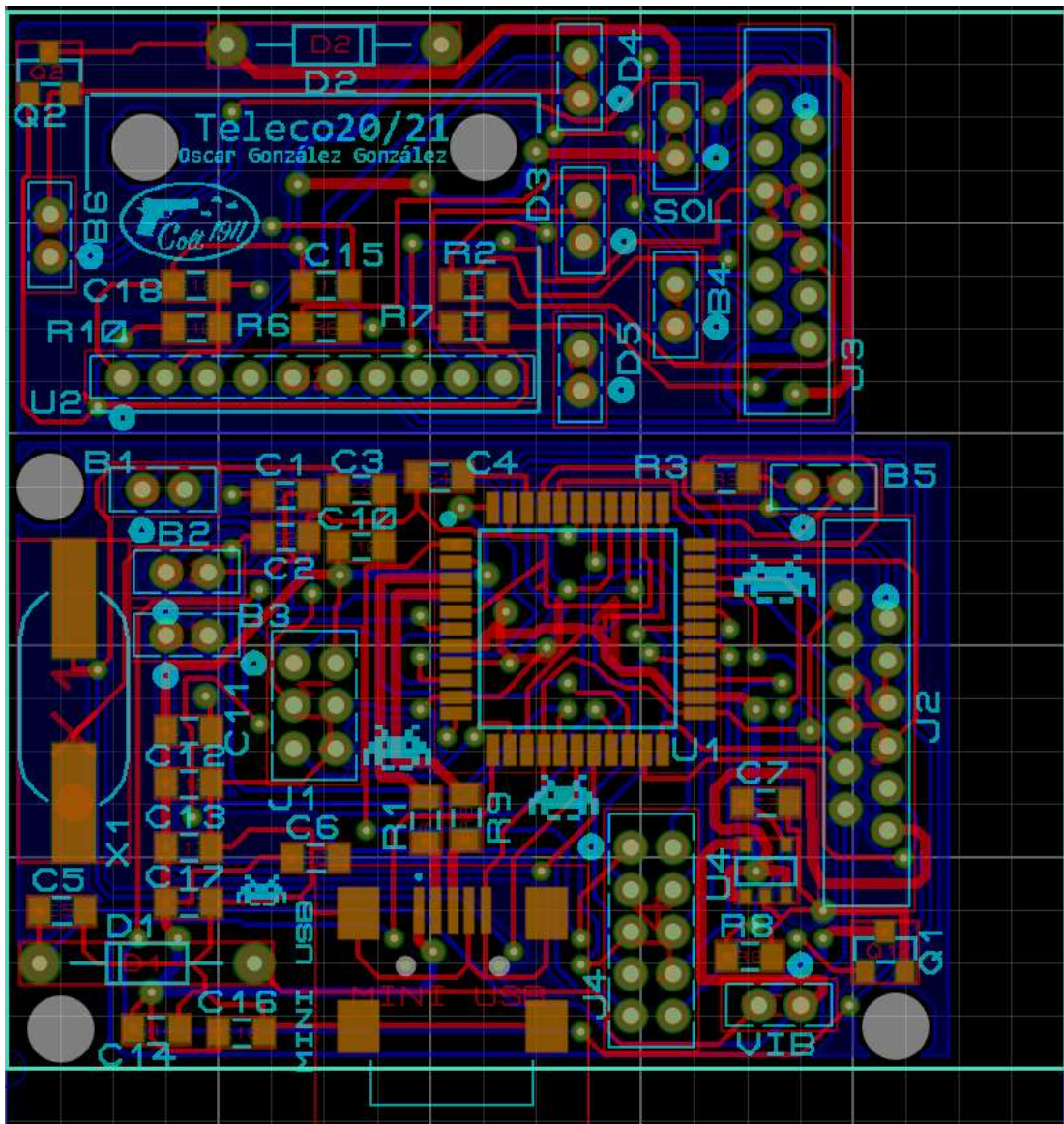
De cara a la adquisición de la implementación física de la placa base, se ha recurrido a la construcción de la misma mediante la empresa PCWay [34]. Para ello, ha sido necesario aportar los ficheros *gerber* o “.GBR”. Con la finalidad de obtener los ficheros *gerber* necesarios para la fabricación de la PCB, se ha recurrido al programa Proteus, en concreto en la pestaña del menú Output>Generate Gerber/Excellon Files. En la **Figura 41** se muestra la ventana emergente con la configuración para la generación de los ficheros *gerber* en el programa Proteus.



**Figura 41.** Configuración para la generación de ficheros gerber mediante el programa Proteus 8.9

Se ha seleccionado el formato RS274X por ser el más antiguo, con el objetivo de asegurar la mayor compatibilidad en el proceso. Es necesario destacar que con anterioridad a la generación de los ficheros *gerber*, se ha procedido a realizar una comprobación de preproducción en el programa Proteus 8.9 tratando de verificar que no existían errores de diseño en la PCB.

Una vez establecida la configuración de los ficheros de salida, el propio programa permite visualizar de forma gráfica la configuración para la creación de la PCB, esta información se muestra en la *Figura 42*.



*Figura 42.* Salida gráfica en el programa Proteus 8.9 tras la generación de los ficheros *gerber*.

Una vez generados los ficheros *gerber* y agrupados en un archivo comprimido en formato “.ZIP”, se ha realizado la solicitud de fabricación en PCBWay. Tras la verificación de los ficheros por parte de la empresa, se ha dado el visto bueno a la información aportada y el pedido se ha aceptado para su fabricación.

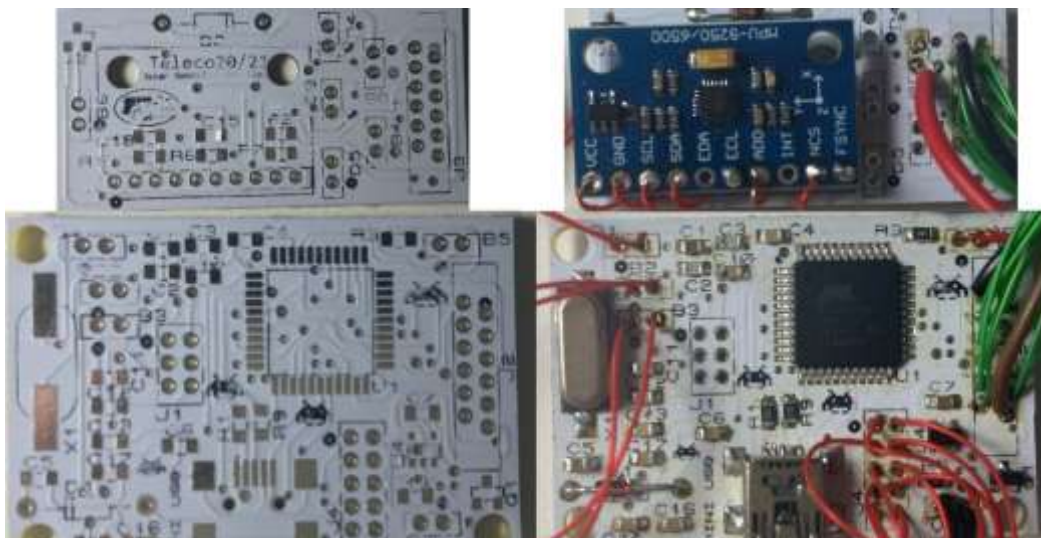
En la sección 1 del anexo que incorpora este documento se incluirán algunos detalles sobre las características de la PCB seleccionadas en PCBWay, con el fin de no sobrecargar la parte principal del documento.

### 3.6 Montaje de componentes

Tras haber recibido todos los elementos físicos que formarán parte del circuito, así como las PCB, se ha procedido a la implementación física del *hardware*.

A la hora de abarcar esta tarea se ha recurrido principalmente a la utilización de un estañador funcionando a 400°C. Además, se ha tratado de llevar a cabo una construcción lo más precisa posible empleando para ello un microscopio. Con el objetivo de facilitar la tarea, también se ha hecho uso de *flux*, permitiendo así evitar la oxidación del estaño en contacto con el aire y mejorando la calidad de las soldaduras. Inicialmente ha sido necesario realizar la fragmentación de la PCB recibida mediante una cizalla, obteniendo así dos partes diferenciadas que permitirán redistribuir de forma sencilla el circuito desarrollado en el interior de la carcasa plástica.

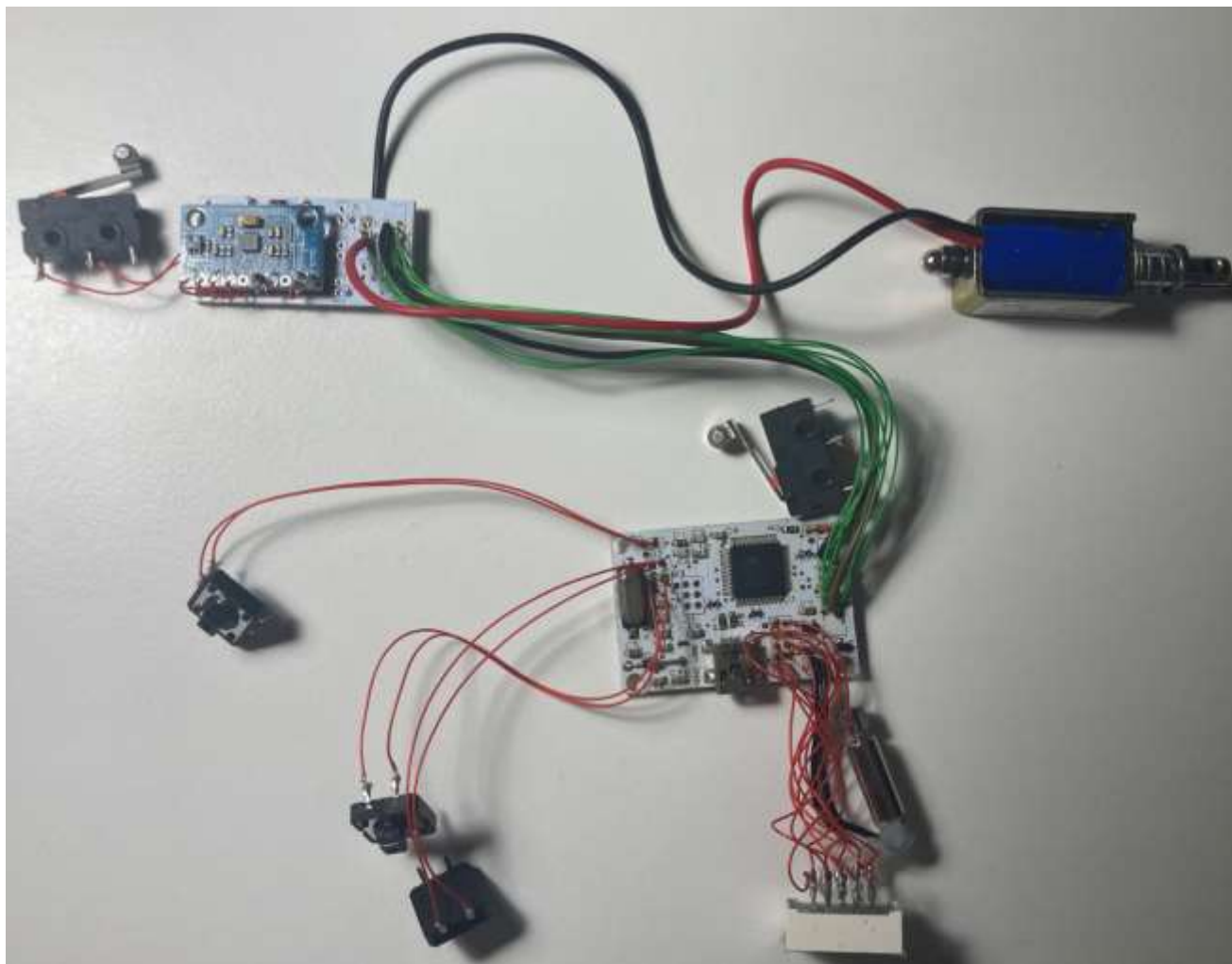
En la **Figura 43** se muestra una imagen comparativa entre el estado inicial y final de las PCB tras la incorporación de sus distintos elementos.



**Figura 43.** Comparativa tras realización del montaje de componentes en la PCB.



Finalmente, puede observarse una fotografía a tamaño completo de todos los componentes del circuito tras el montaje en la siguiente figura.



*Figura 44. Circuito electrónico completo tras el montaje de sus componentes.*

## 4. SOFTWARE

A lo largo de esta sección se tratarán los temas relacionados con la implementación del código fuente en el microcontrolador utilizado, así como los fundamentos matemáticos teóricos en los que se basa el código. Además, se explicarán los pasos llevados a cabo a la hora de preparar el entorno de trabajo.

### 4.1 Instalación del gestor de arranque

Una vez realizada la implementación circuital completa, será necesario dotar a nuestro conjunto electrónico de un gestor de arranque que permita su iteración a través del puerto mini USB incorporado. El gestor de arranque, o *bootloader*, permitirá que el microcontrolador de nuestro circuito sea reconocible durante su puesta en marcha. Para aclarar la necesidad de la instalación de un gestor de arranque deberemos esclarecer las partes de código que incorpora nuestro microcontrolador.

El microcontrolador Atmega32u4 adquirido, cuenta con un *firmware* de fábrica, el cual se encarga de las tareas básicas de funcionamiento del mismo, tales como comprobación del reloj, gestión básica del consumo eléctrico, etc [15]. Este *firmware* es además modificable.

En cuanto al *bootloader* por defecto, proveniente del microcontrolador adquirido, nos encontramos con que tras haber realizado el montaje y conectar mediante USB nuestra placa, no era detectada, lo que nos da a entender que el *bootloader* preinstalado no cuenta con las funcionalidades USB. Por el contrario, según se cita en su hoja de datos “*The On-chip ISP Flash allows the program memory to be reprogrammed in-system through an SPI serial interface, by a conventional nonvolatile memory programmer, or by an On-chip Boot program running on the AVR core.*”[15]. Esto nos permitirá modificar el *bootloader* existente mediante la conexión SPI.

A la hora de realizar la instalación de un nuevo *bootloader* se ha procedido a emplear el conector ICSP que se adelantaba en el apartado de ISP correspondiente con la sección 3.2.2.7 de este documento.

Dado que nuestro circuito no era reconocible mediante USB, se ha recurrido a emplear un programador USBasp. El programador USBasp consiste en un circuito integrado que cuenta con un microcontrolador Atmel AVR (perteneciente a la misma familia de microcontroladores que el empleado en nuestro circuito), así como un conector ISP y

otros elementos pasivos necesarios como condensadores y resistencias. Su principal característica es que contiene un *firmware* preinstalado de fábrica compatible con los *drivers* USB, por lo que será reconocido por nuestro ordenador. En la **Figura 45** se muestra una imagen del programador USBasp adquirido para la instalación del *bootloader*.

Tras haber conectado el programador USBasp, se han realizado las conexiones ICSP entre el programador y nuestro circuito, siguiendo el esquema de conexiones expuesto en la



**Figura 45.** Programador USBasp empleado en la instalación del *bootloader*

**Figura 35** perteneciente a la sección 3.2.2.7 de este documento.

Finalmente, siguiendo la guía proporcionada por Arduino para la instalación del *bootloader* [32], se ha seleccionado “USBasp” como programador y se ha procedido a “quemar” el *bootloader* correspondiente con el circuito integrado Arduino Micro, el cual incorpora el mismo microcontrolador empleado en nuestro circuito.

Tras haber realizado la instalación del *bootloader*, nuestro circuito ha sido reconocido por el equipo informático como un Arduino Micro, mediante el conector USB, lo cual nos permite emplear las bibliotecas disponibles para Arduino, así como su IDE.

La utilización del *bootloader* Arduino facilitará el desarrollo del *software* permitiendo emplear distintas bibliotecas aportadas por la comunidad, que de otra manera, sería necesario desarrollar desde cero.

## 4.2 Bibliotecas empleadas

A lo largo de este apartado pasarán a describirse las bibliotecas empleadas provenientes del conjunto de bibliotecas compatibles con Arduino. Para ello, se expondrá la argumentación relativa a la decisión de incorporar algunas de ellas, además de esclarecer de forma básica el funcionamiento de las mismas.

### 4.2.1 AbsMouse.h

Inicialmente nos centraremos en la implementación del movimiento del ratón. Durante el estudio general del sistema de este documento se discutió la necesidad de implementar un microcontrolador que fuera compatible con USB HID, en este punto podrá dilucidarse la validez de la conclusión anteriormente aportada.

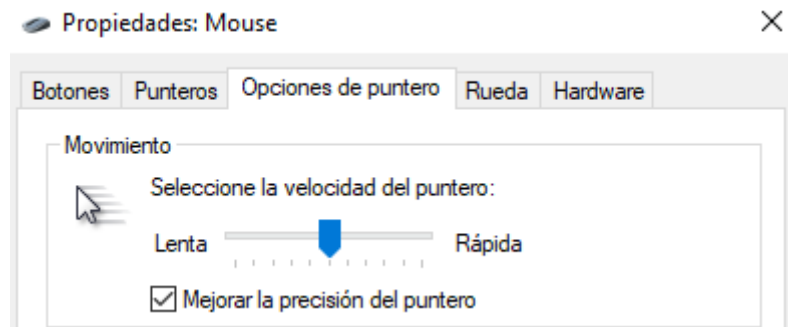
En un momento inicial se recurrió a la biblioteca nativa “Mouse.h” que Arduino presenta en su propia documentación [35]. Sin embargo, esta decisión quedó descartada tras comprobar que el posicionamiento relativo del cursor aportado por esta biblioteca no era viable. Tal y cómo se cita en la descripción de su funcionamiento “*When updating the cursor position, it is always relative to the cursor’s previous location.*”[35].

La utilización de un posicionamiento relativo del cursor aportaba una amplia compatibilidad, por el contrario, presentaba grandes retos debido a los desajustes subyacentes en la configuración del *mouse* de cara a distintos sistemas operativos.

A la hora de descartar la biblioteca nativa proporcionada por Arduino se han tenido en cuenta los siguientes inconvenientes:

- La mayoría de sistemas operativos presentan una aceleración en el cursor que se encarga de determinar la posición, esta aceleración suele estar preconfigurada por defecto. El problema en su implementación, reside en que la distancia recorrida por el puntero depende de la velocidad a la que se realice el movimiento e.g. si se realiza un movimiento horizontal de 1cm en un ratón convencional, el cursor se moverá distancias diferentes dependiendo de la velocidad a la que se recorra ese centímetro.

Esta característica producía que los movimientos en pantalla no fueran precisos de cara a nuestro dispositivo apuntador, con lo cual recaería en tarea del usuario deshabilitar esta opción en el O.S. y la forma de proceder en cada sistema varía de forma amplia. En la **Figura 46** se muestra cómo la opción de aceleración (denominada en Windows 10 como “Mejora la precisión del puntero”) viene marcada por defecto.



**Figura 46.** Aceleración del mouse latente en el O.S. Windows 10.

A la hora de realizar la implementación con la biblioteca “*Mouse.h*” las funciones aportadas para el movimiento, en este caso “*Mouse.move()*” producían que a mayor desplazamiento en pantalla se recorriera una mayor distancia, es decir, la suma de varios desplazamientos en pantalla no era equivalente a un desplazamiento de igual magnitud.

- El segundo inconveniente de utilizar posicionamiento relativo reside en determinar el centro de la pantalla. A la hora de realizar la calibración de nuestro producto será necesario partir desde algún punto, típicamente el centro de la pantalla, sin embargo, es complejo determinar el centro de una pantalla rectangular mediante posicionamiento relativo, dado que no conocemos la posición desde la que se parte.

Durante la implementación del algoritmo de posicionamiento se consiguió de forma exitosa colocar el cursor en el centro de la pantalla, a pesar de esto, la idea quedó descartada, debido a que si quisiéramos implementar este proceso en una pantalla con una resolución distinta el algoritmo no sería válido.

- Como argumento final, citar que, la biblioteca “*Mouse.h*” en sus funciones destinadas al movimiento, únicamente incorpora 8 bit en sus parámetros, permitiendo movimientos máximos de entre -127 y 127. Este rango es insuficiente a la hora de realizar cambios bruscos en pantallas donde la resolución sea superior a 255x255 píxeles.

Como consecuencia de estos inconvenientes, se ha recurrido a un posicionamiento absoluto que, a pesar de poseer un desarrollo de programación con mayor complejidad y gozar de una menor compatibilidad con aplicaciones y dispositivos, presentaba una mayor precisión en el movimiento.

Además de necesitar un posicionamiento relativo en el dispositivo señalador, se hacía necesaria la utilización de más bit en los parámetros destinados a determinar el valor del movimiento.

Para poder modificar estos valores sería necesario modificar el descriptor USB, más concretamente, modificar el fichero de código “*Mouse.cpp*” en el que se definen las características de funcionamiento para así poder incorporar un posicionamiento absoluto y más bit en los parámetros de las funciones encargadas del movimiento. Por suerte, debido a la amplia comunidad con la que cuenta Arduino ha sido posible encontrar una biblioteca que incorporaba las modificaciones buscadas, esta biblioteca se denomina “*AbsMouse.h*” y es una aportación independiente de uno de los usuarios de Arduino. Su implementación puede consultarse en el repositorio de código GitHub [36].

Consultando el código fuente del descriptor USB HID para el *mouse*, nos encontramos con que se incorporan 16 bit para el posicionamiento del cursor en pantalla en los ejes x e y tal y cómo se muestra en la **Figura 47**, quedando reservados 15 de ellos para el desplazamiento (valores de 0 a 32767). En esta figura puede apreciarse cómo se incorpora a mayores un posicionamiento absoluto.

```
0x09, 0x30, // Usage (X)
0x09, 0x31, // Usage (Y)
0x16, 0x00, 0x00, // Logical Minimum (0)
0x26, 0xFF, 0x7F, // Logical Maximum (32767)
0x36, 0x00, 0x00, // Physical Minimum (0)
0x46, 0xFF, 0x7F, // Physical Maximum (32767)
0x75, 0x10, // Report Size (16)
0x95, 0x02, // Report Count (2)
0x81, 0x02, // Input (Data,Var,Abs,No Wrap,Linear,Preferred State,No Null Position)
```

**Figura 47.** Código fuente del fichero “*AbsMouse.cpp*” relativo a la utilización de 16 bits en el posicionamiento y a un posicionamiento absoluto. [36]

Otras características de interés que incorpora esta biblioteca, se observan en la **Figura 48**, donde se muestran los métodos “init” y “move”, los cuales permiten inicializar la biblioteca estableciendo la resolución de la pantalla y se calculan de forma automática los valores máximos del movimiento en función de la resolución.

```

void AbsMouse_::init(uint16_t width, uint16_t height, bool autoReport)
{
    _width = width;
    _height = height;
    _autoReport = autoReport;
}

void AbsMouse_::move(uint16_t x, uint16_t y)
{
    _x = (uint16_t) ((327671 * ((uint32_t) x)) / _width);
    _y = (uint16_t) ((327671 * ((uint32_t) y)) / _height);

    if (_autoReport) {
        report();
    }
}

```

**Figura 48.** Código fuente del fichero "AbsMouse.cpp" relativo a los métodos de inicialización y movimiento del cursor. [36]

Esta biblioteca nos permitirá paliar los problemas hallados en el código nativo proporcionado por Arduino, además de incorporar otros métodos alternativos para los botones del ratón tales como “press” o “release”.

#### 4.2.2 Keyboard.h

Las bibliotecas nativas de Arduino presentan la posibilidad de incorporar pulsaciones de teclado. De manera que emplearemos esta biblioteca con la finalidad de que los botones sean reconocidos por los distintos sistemas operativos como si se tratase de teclas de un teclado convencional. En la referencia de funciones de Arduino se especifican los distintos métodos disponibles para esta biblioteca[37].

#### 4.2.3 MPU9250.h

A la hora de realizar la programación relativa al sensor MEMS incorporado, se ha recurrido a la biblioteca “MPU9250.h” concebida para sistemas de vuelo eficaces. La decisión que ha llevado a elegir esta biblioteca se fundamenta en su amplia utilización, su documentación avanzada, las posibilidades de modificación y su compatibilidad con el protocolo SPI.

Además de esto, la biblioteca “MPU9250.h” se fundamenta en otras bibliotecas orientadas a IMU (*Inertial Measurement Unit*) con un recorrido de desarrollo que goza de gran solidez y cuenta con datos que se ajustan a los parámetros de interés que caracterizaremos en nuestros sensores, tal y como se muestra en la **Figura 49**. Según se cita en la descripción de la biblioteca “*This driver conforms to the IMU interface*” [38]. También hemos de añadir que su elección ha venido determinada por haber sido desarrollada de manera específica para nuestro conjunto de sensores “*This library communicates with InvenSense MPU-9250 and MPU-9255 Inertial Measurement Units (IMUs).*”[38].

Name	Description
bool new_imu_data	Whether new accel and gyro data was read
bool new_mag_data	Whether new mag data was read
bool imu_healthy	Whether the accel and gyro are healthy
bool mag_healthy	Whether the mag is healthy
float die_temp_c	The IMU die temperature, C
float accel_mps2[3]	The 3-axis accel data, m/s/s
float gyro_radps[3]	The 3-axis gyro data, rad/s
float mag_ut[3]	The 3-axis mag data, uT

**Figura 49.** Campos de datos pertenecientes a la estructura *ImuData* que recoge los datos del sensor.[39]

En la definición de la biblioteca “imu” [39] pueden consultarse con mayor detalle las clases, métodos y atributos empleados.

Siguiendo con la argumentación acerca de la utilización de la biblioteca “MPU9250.h”, hemos de explicar que, además de facilitar la utilización de los datos de interés, permite configurar los rangos de escala completos de los sensores ya sea del acelerómetro, giroscopio y magnetómetro, así como otras configuraciones importantes tales que el ancho de banda.



Continuando con las características de esta biblioteca, cabe destacar que uno de los factores determinantes ha sido la compatibilidad con SPI, la cual muchas otras no implementan de una forma estable. En la **Figura 50** se muestra la compatibilidad con la clase SPI que permite iniciar de forma sencilla la comunicación mediante este protocolo a través del pin de selección de chip.

```
MPU9250::MPU9250(SPIClass &bus, uint8_t csPin){
    _spi = &bus; // SPI bus
    _csPin = csPin; // chip select pin
    _useSPI = true; // set to use SPI
}
```

**Figura 50.** Objeto MPU9250, entrada al bus SPI y pin de selección de chip del fichero MPU9250.cpp

Finalizando con las características principales de esta biblioteca, las cuales aumentan el interés de elección de la misma, se muestra en la **Figura 51** cómo se implementa la lectura y transformación de los datos a unidades métricas con las que podemos trabajar provenientes de la lectura de los sensores, dotándonos con salidas de datos en  $\frac{m}{s^2}$  para el acelerómetro,  $\frac{rad}{s}$  para el giroscopio y  $\mu T$  en el caso de los datos del magnetómetro.

```
// transform and convert to float values
_ax = (((float)(tX[0]*_axcounts + tX[1]*_aycounts + tX[2]*_azcounts) * _accelScale) - _axb)*_axs;
_ay = (((float)(tY[0]*_axcounts + tY[1]*_aycounts + tY[2]*_azcounts) * _accelScale) - _ayb)*_ays;
_az = (((float)(tZ[0]*_axcounts + tZ[1]*_aycounts + tZ[2]*_azcounts) * _accelScale) - _azb)*_azs;
_gx = ((float)(tX[0]*_gxcounts + tX[1]*_gycounts + tX[2]*_gzcounts) * _gyroScale) - _gxb;
_gy = ((float)(tY[0]*_gxcounts + tY[1]*_gycounts + tY[2]*_gzcounts) * _gyroScale) - _gyb;
_gz = ((float)(tZ[0]*_gxcounts + tZ[1]*_gycounts + tZ[2]*_gzcounts) * _gyroScale) - _gzb;
_hx = (((float)(_hxcounts) * _magScaleX) - _hxb)*_hxs;
_hy = (((float)(_hycounts) * _magScaleY) - _hyb)*_hys;
_hz = (((float)(_hzcounts) * _magScaleZ) - _hzb)*_hzs;
_t = (((float) _tcounts) - _tempOffset)/_tempScale) + _tempOffset;
```

**Figura 51.** Lectura y conversión de los datos en unidades de ingeniería del fichero MPU9250.cpp

#### 4.2.4 SevenSeg.h

A la hora de realizar la configuración del *display* LED de siete segmentos, se ha procedido a utilizar la biblioteca “SevenSeg.h”[40] por ser una de las bibliotecas mejor documentadas en cuanto a su funcionamiento y ser compatible con *displays* de siete segmentos y dos dígitos que cuenten con conexiones compartidas entre sus LED. De esta forma nos estaremos ahorrando la programación desde cero sobre los ciclos de reloj empleados en mostrar diferentes “caracteres” para cada uno de los dígitos, engañando al ojo humano.

Además de ser una biblioteca de sencilla implementación, cuenta con la posibilidad de escribir caracteres alfanuméricos y permite una programación de bajo nivel si fuese necesario.

Su documentación es sólida y cuenta con un manual en el que se explican las distintas funcionalidades de forma exhaustiva [41].

#### **4.2.5 SoftwareReset.h**

Concluyendo con las bibliotecas empleadas, se ha incorporado la biblioteca “SoftwareReset.h” [42] con el fin de poder realizar un “reinicio” del microcontrolador. De esta forma, se permitirá al usuario modificar los parámetros del cargador y otras configuraciones, realizadas únicamente al inicio, sin la necesidad de desconectar y conectar nuevamente el dispositivo.

La implementación de esta biblioteca se fundamenta en realizar un bucle infinito que fuerce el bloqueo del programa, estableciendo el WDT (*WatchDog Timer*) en el mínimo valor, siendo este de 15ms mediante la el comando `wdt_enable(WDTO_15MS)`. Consiguiendo reiniciar el microcontrolador mediante *software*.

Durante la sección 4.2 de este documento se han descrito las características generales de las bibliotecas empleadas, sin embargo, cabe destacar que los métodos empleados y la descripción del código de estas bibliotecas, así como sus funcionalidades, serán descritas mediante comentarios junto al código proporcionado en el anexo de este documento con la finalidad de no sobrecargar el grueso principal del mismo.

### **4.3 Desarrollo teórico del código implementado**

Continuando con el desarrollo del *software* de nuestro producto, se pasarán a explicar, en este apartado, los fundamentos matemáticos empleados en la determinación de la posición del puntero en la pantalla. A modo orientativo se realizará inicialmente una breve explicación general sobre el código fuente del programa.

#### **4.3.1 Estructura general del código fuente**

El código fuente del programa está realizado en el lenguaje de programación C++ relativo a Arduino.

Los programas desarrollados con el IDE Arduino cuentan con dos partes principales: la función `setup()` y la función `loop()`. Estas partes están destinadas a la inicialización del programa y a el bucle de ejecución respectivamente.

El código implementado se inicia con la incorporación de las bibliotecas citadas en el apartado 4.2 de este documento. Posteriormente se definen las *macros* referentes a los pines físicos del microcontrolador que tendrán distintos papeles, principalmente ligadas a los botones, el *display* LED y los actuadores.

Tras haber definido las macros, se crean las variables globales que formarán parte del programa relativas a los cálculos que determinarán la posición. Además de esto se incorporan instancias de objetos propios de las bibliotecas “SevenSeg.h” y “MPU9250.h”.

#### **4.3.1.1 Setup**

Esta función es la primera de las dos principales, en lo que al IDE Arduino se refiere. En nuestro caso se emplea para llevar a cabo las tareas de configuración inicial, inicialización de valores relativos a los objetos de las bibliotecas incorporadas y otros aspectos como la calibración.

El código de esta función consta, en primer lugar, con un menú que permitirá ajustar diferentes configuraciones tales que el tamaño del cargador, la resolución, la distancia del jugador a la pantalla y otros aspectos cómo las pulgadas de la pantalla empleada. Los detalles de los menús y las posibilidades de configuración se aportarán en el anexo de este documento en forma de manual de usuario.

En la parte final de esta función se realizan unos cálculos matemáticos primarios en función de las configuraciones del menú. De esta forma, estaremos preparando los valores de las variables globales para su interpretación en la función `loop()`.

Para concluir, se inicializan los objetos de la biblioteca “MPU9250.h” realizando así la calibración de los sensores y estableciendo los rangos de escala completos.

#### **4.3.1.2 Loop**

La segunda función principal del código se encargará de ejecutar el código de su interior de manera cíclica una y otra vez.

A su inicio, se establece la lectura de los sensores para cada uno de los ciclos de ejecución del bucle. Así mismo, se incorpora el algoritmo que determinará la posición en pantalla, teniendo en cuenta los valores procedentes de la lectura del sensor y las variables globales preparadas tras la configuración en la función `setup()`.

Además de esto, se crean los comportamientos de los botones en función de posibles acontecimientos de accionamiento.

Concluyendo con la segunda función principal del programa, se establece una combinación de botones que permita realizar un reinicio por *software* empleando la biblioteca “SoftwareReset.h”.

### **4.3.2 Desarrollo del algoritmo de posicionamiento**

Una vez aclaradas de manera general las secciones del código fuente, nos centraremos en describir el algoritmo matemático empleado. Este algoritmo se utiliza para determinar las posiciones en pantalla en función de los valores de los sensores.

Inicialmente se trató de determinar la posición de los sensores empleando una combinación de datos provenientes tanto del magnetómetro y acelerómetro como del giroscopio. Sin embargo, debido a la baja precisión de los sensores de aceleración y magnéticos y teniendo en cuenta sus valores de referencia absolutos, se ha decidido trabajar únicamente con los datos provenientes del giroscopio.

La decisión de trabajar de forma única con las lecturas de datos asociadas al giróscopo viene propiciada por los errores de cuantificación, de los algoritmos probados. El error de cuantificación en el caso de los dos sensores restantes era incompatible con la precisión buscada. El problema principal reside en que las pantallas actuales cuentan con resoluciones elevadas.

Como consecuencia, al tratar de implementar algoritmos basados en el acelerómetro, uno de los sensores más capaces de determinar la posición exacta, los movimientos del puntero en la pantalla resultaban en saltos escalonados. A este problema, se le suma el hecho de que el acelerómetro posee un sistema de referencia fija, midiendo la posición absoluta, al contrario que el giróscopo que nos permite medir los incrementos relativos.

Tras haber realizado múltiples pruebas *software* aplicando distintos algoritmos, incluso tratando de aminorar el error de cuantificación de los datos provenientes del acelerómetro,

se ha determinado que la implementación que aportaba mejores sensaciones de usuario era la que únicamente se basaba en el giroscopio.

Aclaradas las decisiones del tratamiento de los datos se pasarán a relatar los algoritmos introducidos desde un punto de vista matemático.

#### 4.3.2.1 Fundamentos matemáticos

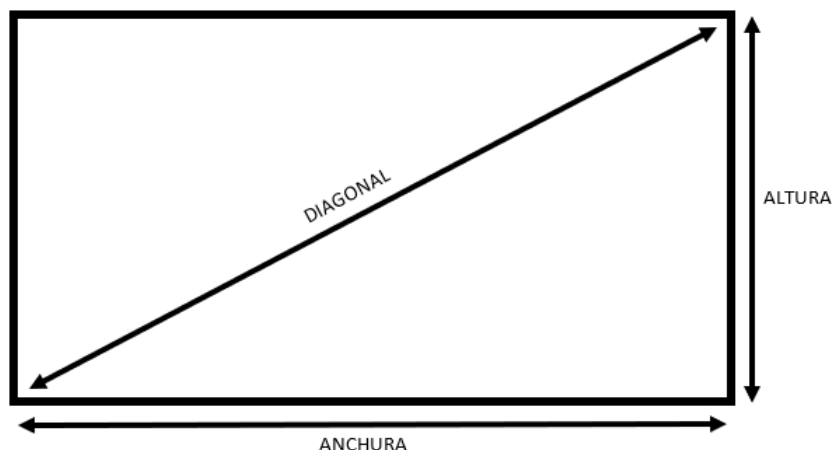
A la hora de abordar el problema y teniendo en cuenta que únicamente se ha empleado el giroscopio, se han tratado de definir una serie de variables (las cuales son configurables) que permitan acotar el problema.

En primer lugar, trataremos de calcular las proporciones de la pantalla utilizada, para ello debemos aclarar que únicamente trabajaremos con dos relaciones de aspecto distintas, siendo estas 16:9 y 21:9.

Nuestro producto será compatible con pantallas de las siguientes resoluciones:

- “HD” (*High Definition*) 1366x768(16:9)
- FHD (*Full HD*) 1920x1080 (16:9)
- QHD (*Quad HD*) 2K 2560x1440(16:9)
- UW (*UltraWide*) 3440x1440 (21:9)
- UHD (*Ultra HD*) "4K" 3840x2160 (16:9)

Una vez el usuario haya seleccionado la resolución, conoceremos la relación de aspecto de manera inmediata. El usuario deberá seleccionar también el tamaño de la pantalla, indicando para ello las pulgadas.



*Figura 52. Ayuda visual para el planteamiento matemático.*

Nuestro objetivo será determinar las dimensiones reales de la pantalla en centímetros. Supondremos que estamos trabajando con una pantalla plana.

Obtenidos los valores de la resolución y las pulgadas podremos calcular la altura y la anchura de la pantalla. En primer lugar, realizaremos una conversión de unidades de pulgadas a centímetros:  $Centímetros = Pulgadas * 2.54$ , con la finalidad de obtener el valor de la diagonal. Posteriormente aplicaremos el teorema de Pitágoras para calcular la anchura y altura:

$$Relación\ de\ aspecto = \frac{16}{9} \text{ ó } \frac{21}{9} = \frac{Numerador}{Denominador}$$

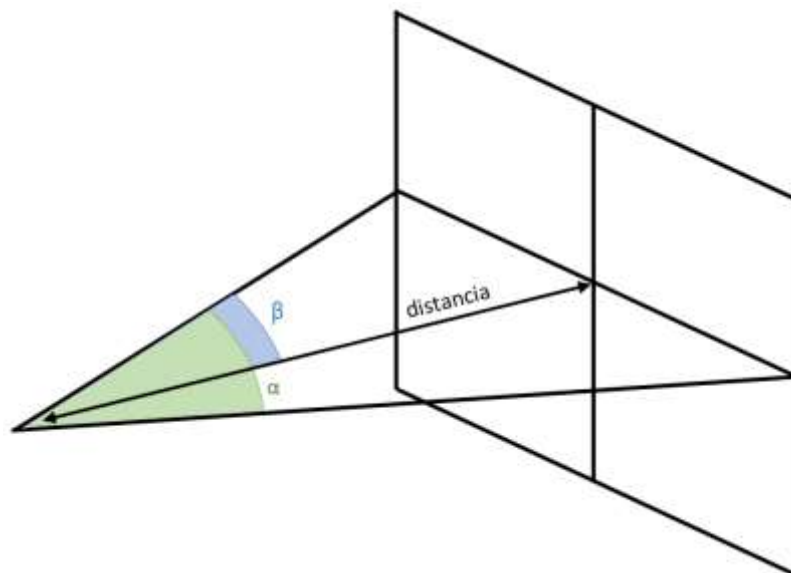
$$Pitágoras: diagonal^2 = altura^2 + anchura^2$$

$$Diagonal^2 = (Numerador * c)^2 + (Denominador * c)^2$$

$$c = \sqrt{\frac{Diagonal^2}{Numerador^2 + Denominador^2}}$$

Tras haber aplicado Pitágoras ya tendremos las dimensiones de la pantalla, la **Figura 52** se aporta como ayuda visual al planteamiento matemático.

Una vez calculadas las dimensiones de la pantalla y dado que la salida del giroscopio viene proporcionada en rad/s, tendremos que calcular el ángulo de barrido de las dimensiones de anchura y altura en función de la distancia del jugador a la pantalla. En la **Figura 53** se muestran los parámetros a calcular.



**Figura 53.** Ángulos de barrido relativos a la anchura en función de la distancia.

La distancia será un parámetro configurable por parte del usuario, con lo cual, poseyendo la distancia y la anchura, podremos calcular la hipotenusa de uno de los dos triángulos

rectángulos mostrados en la **Figura 53**, y con ello el ángulo  $\beta$ , que multiplicado por dos nos aportará el ángulo de barrido completo  $\alpha$ .

$$Hipotenusa = \sqrt{\left(\frac{Anchura}{2}\right)^2 + Distancia^2}$$

$$\alpha = 2 * \beta = 2 * \arcsen\left(\frac{\frac{Anchura}{2}}{Hipotenusa}\right) * \frac{\pi}{180} \text{ rad}$$

A la hora de calcular el ángulo de barrido en la altura, se procederá de manera similar.

Llegados a este punto, hemos podido obtener el ángulo de barrido en radianes, sin embargo, el giroscopio nos aporta los datos en rad/s. Con lo cual, será necesario calcular el tiempo que tarda el programa en llegar a la lectura de los datos del giroscopio con la finalidad de obtener una referencia en rad/s. Esta tarea se ha realizado mediante la práctica, haciendo uso de la función `millis()` y efectuando una resta entre el contador del tiempo de ejecución al inicio del bucle y al final de la lectura de los datos.

Ahora nuestros datos de referencia estarán en rad/s, donde los radianes son los calculados respecto a las dimensiones de la pantalla y los segundos es el tiempo que tarda el programa en llegar hasta la lectura del sensor.

$$rad/s_{referencia} = \frac{\text{Ángulo de barrido}}{\text{Tiempo del bucle en consultar sensor}} \rightarrow rad/s_{sensor}$$

Teniendo los datos en las mismas unidades, podremos realizar una cuantificación de los datos del sensor en función de la resolución. Al inicializar `AbsMouse` e indicar la resolución, se escalan los valores de coordenadas máximos en función de la resolución de la pantalla. De esta manera, los máximos puntos en pantalla quedan definidos por los valores máximos de la resolución vertical y horizontal.

Al realizar el movimiento en pantalla, el cursor se moverá al píxel exacto, correspondiente con la cuantificación empleada. Con el fin de esclarecer el proceso nos centraremos en una línea de código. Únicamente trabajaremos con el ángulo de barrido horizontal, dado que en el vertical la matemática aplicada es idéntica.

```
AbsMouse.move((((angRad_state[0]-
IMU.getGyroY_rads())*(resolucion[0]/angRad[0]))),mouse_y_be
fore);
```

En la anterior línea de código se expone la idea matemática principal del funcionamiento. Habiendo transformado los radianes máximos calculados teóricamente, se establece, en un inicio, el estado de los rad/s en el centro de la pantalla. A esa posición en rad/s se le restan los rad/s captados por el sensor (el signo menos proviene de la orientación del eje según las bibliotecas). Una vez obtenida la diferencia se realiza una cuantificación en función de la resolución y los rad/s máximos calculados, quedando la siguiente expresión:

$$\begin{aligned}
 \text{Píxel}_{horizontal} &= (\text{rad}/s_{referencia\_actuales} - \text{rad}/s_{variación\_sensor}) \\
 & * \left( \frac{\text{resolución}_{horizontal}}{\text{rad}/s_{referencia\_máximos}} \right)
 \end{aligned}$$

La anterior expresión podría haberse concebido de una forma más intuitiva transformando los rad/s del sensor directamente a radianes. Esta opción ha sido descartada con el fin de reducir la carga computacional en cada lectura del sensor y trabajar con valores fijos.

La expresión alternativa sería:

$$\begin{aligned}
 \text{Píxel}_{horizontal} &= (\text{rad}_{referencia\_actuales} \\
 & - (\text{rad}/s_{variación\_sensor} * \text{Tiempo del bucle en consultar sensor})) \\
 & * \left( \frac{\text{resolución}_{horizontal}}{\text{rad}_{referencia\_máximos}} \right)
 \end{aligned}$$



## 5. DISEÑO DE LA CARCASA

A la hora de proceder con el diseño de la carcasa del producto, se ha partido de un diseño que ya poseía la parte externa creada. De manera, que no se ha desperdiciado demasiado tiempo en la realización de la estética exterior, pudiendo así asignar un mayor trabajo en la parte interna, que será la que determinará las funcionalidades del arma y permitirá implementar los componentes seleccionados según sus limitaciones físicas.

Inicialmente se buscó un diseño atractivo. Como la finalidad del proyecto es realizar un dispositivo electrónico que sea capaz de emular al máximo posible las sensaciones reales de empuñar un arma, se trató de encontrar un diseño basado en un arma real. Al querer utilizarse en entornos recreativos, el diseño del arma debería de ser un tanto icónico y vistoso. Por lo que se decidió implementar como aspecto exterior el diseño de la pistola Colt 1911.

Esta pistola fue creada en 1911 [43] y es por esto, que cuenta con una historia relativamente larga. Como aliciente, su longeva edad le ha permitido intervenir en múltiples series, películas y videojuegos, constatando así su atractivo icónico y aportando un valor añadido al producto final.

Algunos de estos ejemplos pueden ser su utilización en películas como “Indiana Jones y el Reino de la Calavera de Cristal”, “La Momia”, incluso en películas más recientes como “Wonder Woman” [44]. Además, en el grueso de la industria del videojuego ha sido empleada en diversos títulos a lo largo de los años e.g. “Call of Duty” y sus múltiples versiones, “Battlefield” o “FarCry”, entre otros muchos.

Esta pistola cuenta con seguidores acérrimos que disfrutarían utilizándola en un videojuego, un claro ejemplo de esto puede verse en vídeos como [45] donde algún fan del arma ha decidido realizar una comparación de la misma en su paso por los videojuegos.

Expuesta la propuesta de diseño exterior, en la que se ha tratado de aportar cierto valor añadido al producto, se pasará a describir el desarrollo planteado durante el diseño de la carcasa.

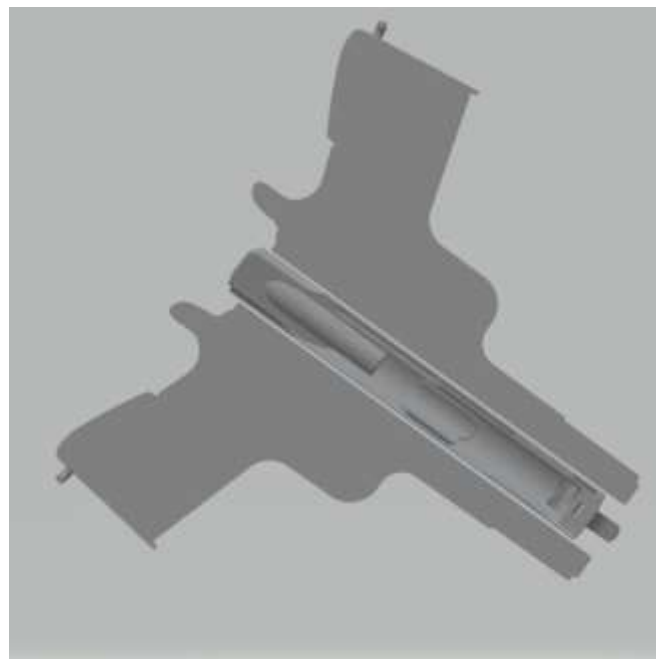
## 5.1 Desarrollo del diseño tridimensional

Al tener definido el concepto de diseño, únicamente era necesario encontrar un desarrollo lo suficientemente avanzado como para poder partir de él. Buscando por la red se encontró un buen diseño que permitiría llevar a cabo la realización de un prototipo de carcasa interior, partiendo de piezas con cierto grado de realismo.

Por lo que, se decidió utilizar el diseño que se muestra en las **Figuras 54 y 55**. Este diseño fue elegido por presentar un tamaño proporcional a la realidad y mostrar un despiece que facilitaría la edición tridimensional.



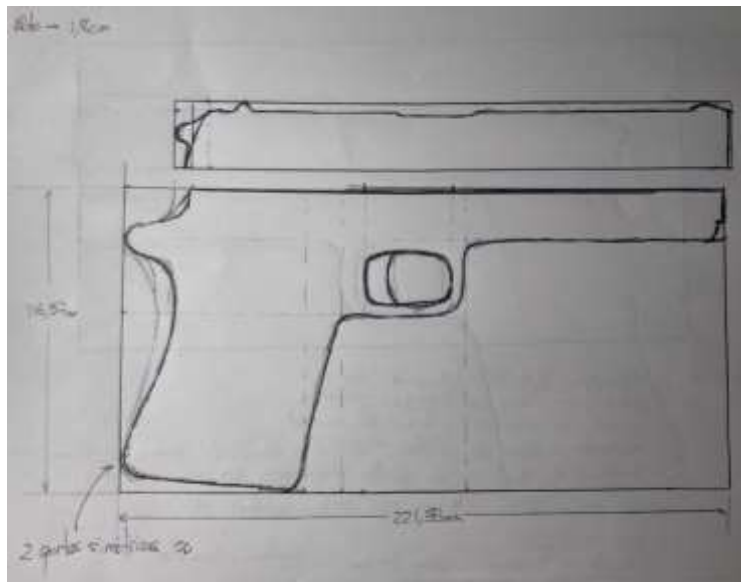
*Figura 55. Diseño de partida, forma exterior.*[46]



*Figura 54. Diseño de partida, forma interior.* [46]

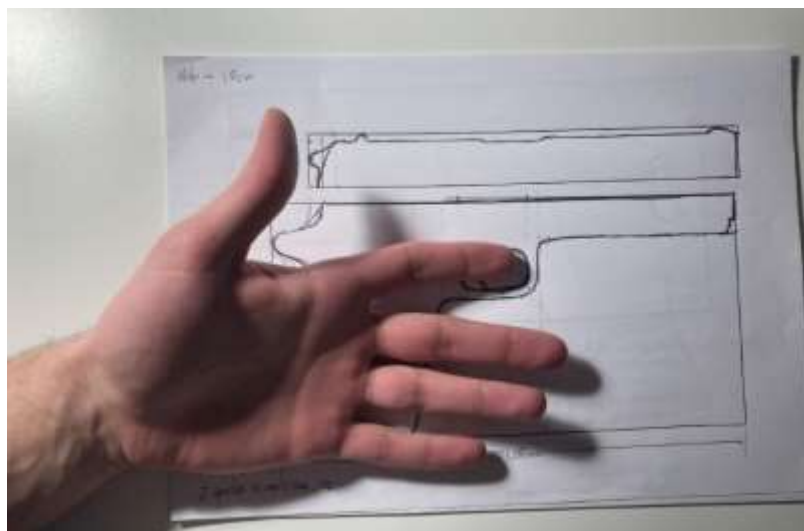
El diseño proviene de la página web Thingiverse, una plataforma donde la comunidad de usuarios aporta diseños tridimensionales orientados a impresión. En concreto, se ha empleado un réplica a tamaño proporcional de la pistola Colt 1911 [46].

Tras haber encontrado un diseño que podría adaptarse a las necesidades del proyecto, se realizó un bosquejo a papel y boli, con el fin de dilucidar si esta opción sería válida por su tamaño. El diseño inicial a escala puede observarse en la **Figura 56**.



**Figura 56.** Bosquejo inicial del diseño a tamaño real.

También se comprobó si sería ergonómicamente viable, o si la proporcionalidad respecto a una mano sería la adecuada. En principio no se realizaron recortes, aunque puede apreciarse una imagen en perspectiva respecto al tamaño de una mano en la **Figura 57**.

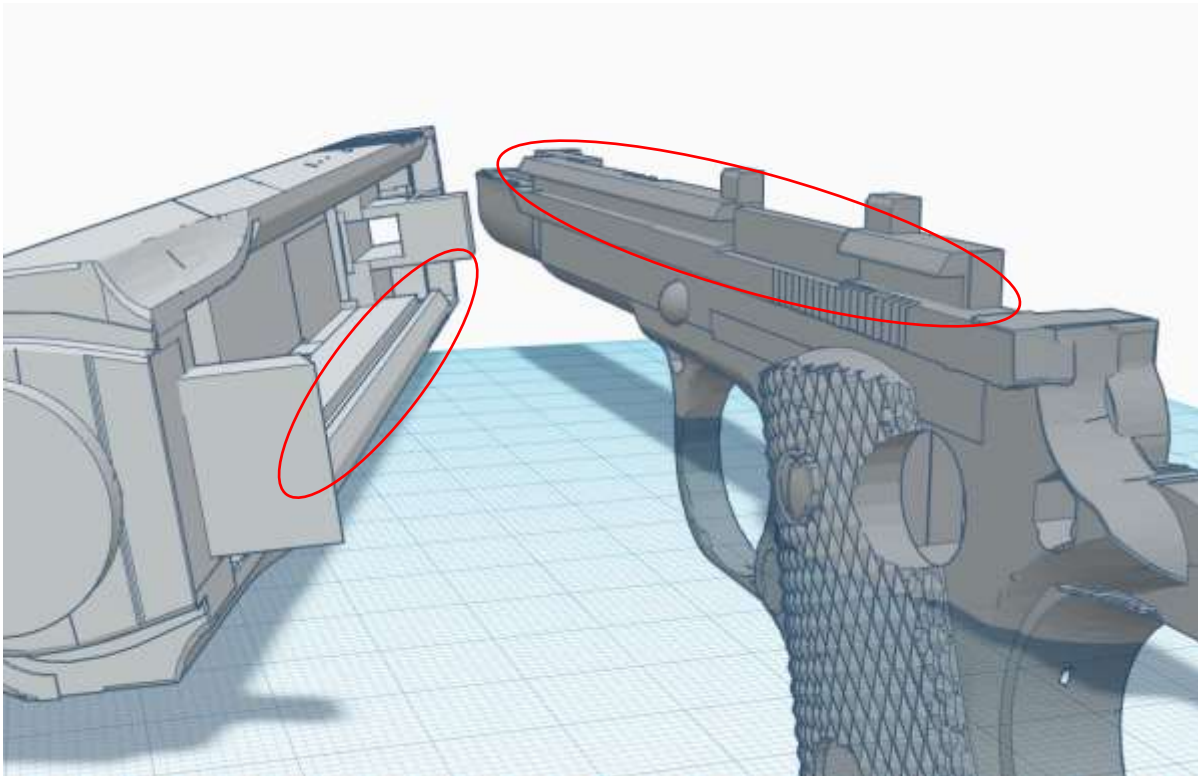


**Figura 57.** Vista en perspectiva del bosquejo frente a una mano.

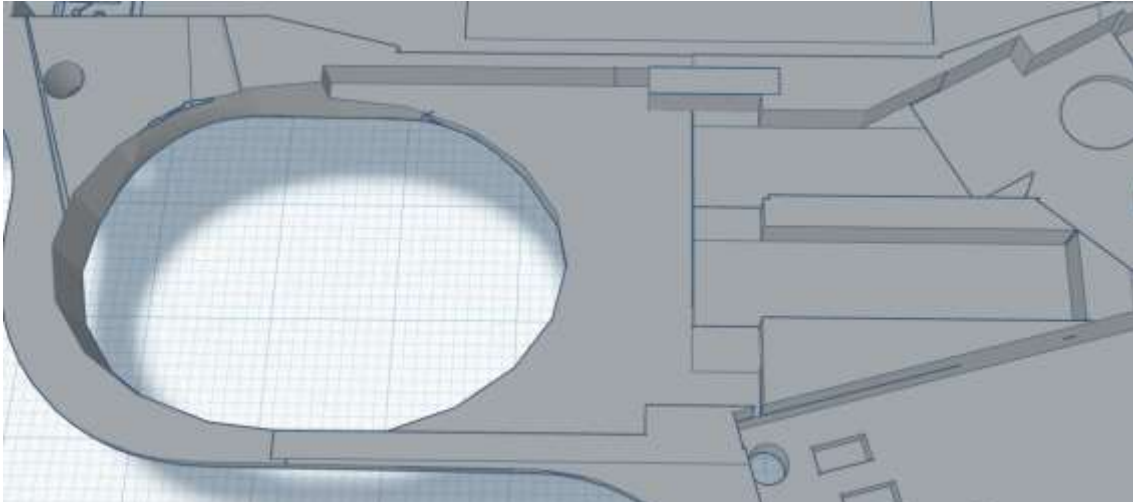
Después de haber comprobado que el diseño podría ser manipulable y que las dimensiones en tamaño real eran las esperadas, se dio paso a la edición del diseño en tres dimensiones. Para ello se utilizó la herramienta *online* Tinkercad[47].

En los inicios, la edición tridimensional se centró en eliminar las partes internas que no serían útiles, como el cañón interno de la corredera, así como a intentar aprovechar la mayor parte del interior del diseño, pero sin restar cualidades de resistencia a la carcasa.

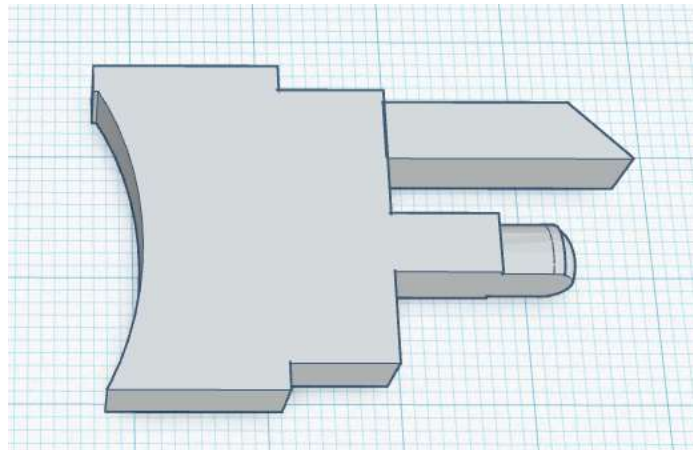
Una vez se había “limpiado” el diseño inicial para poder empezar a operar con él, comenzaron a incorporarse partes que permitían encajar las piezas. Esto fue necesario, dado que el diseño venía pensado para pegar las piezas unas con otras mediante superficies lisas, sin embargo, nosotros necesitaremos encajar las piezas con disposiciones mecánicas. Una de las primeras aportaciones fue la incorporación de railes tanto en la base como en la corredera de la pistola, para así poder encajar las piezas de una forma sólida, facilitando el deslizamiento de la corredera. La implementación de este mecanismo puede observarse en la **Figura 58**.



**Figura 58.** Muestra del sistema de acoplamiento de la carcasa mediante guías.



**Figura 5959.** Posicionamiento y raíles para la incorporación del gatillo.



**Figura 60.** Pieza tridimensional del gatillo.

El funcionamiento del gatillo se basaría en una pieza suelta, de forma que al encajar ambas partes simétricas de la base de la pistola este quedara sujeto, pero pudiendo desplazarse a lo largo de los raíles marcados. El gatillo posee dos salientes, el saliente inferior posee una mitad cilíndrica y una última parte con forma de prisma, esta forma ha sido concebida así con el fin de incorporar un muelle que permita al gatillo volver a su lugar de origen tras ser accionado. Este muelle no se desplazará de su posición gracias a que se ha creado una guía que impedirá movimientos descontrolados.

Por otra parte, el saliente superior mostrado en la **Figura 60**, ha sido concebido con el fin de accionar un interruptor de final de carrera con rodillo y por esto posee una superficie inclinada en su punta, permitiendo así que el rodillo del interruptor se desplace sin problemas a lo largo de esta.

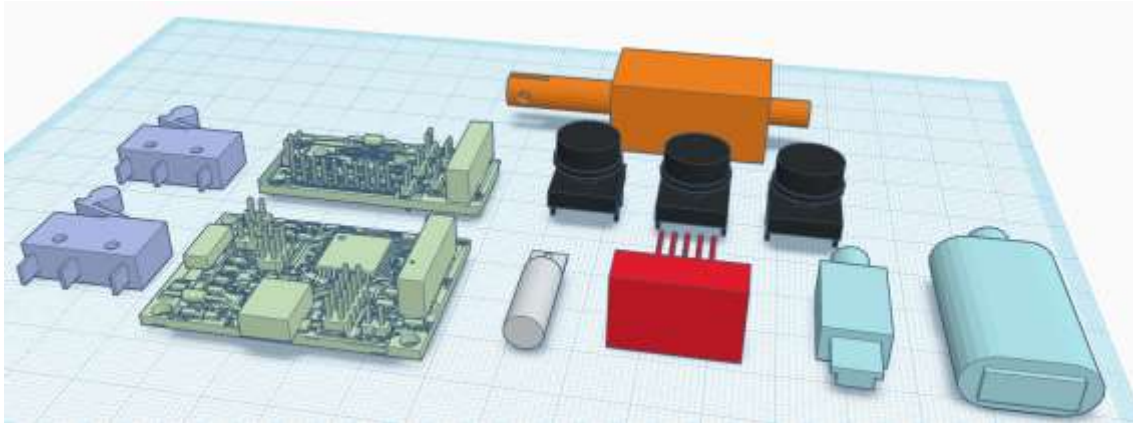
Una vez expuestos estos cambios, pasaremos a comentar las modificaciones llevadas a cabo en la corredera.



*Figura 60. Sección de la base con corredera acoplada.*

Según se muestra en la **Figura 61**, en la corredera se han incorporado dos topes. El posicionado en la parte superior izquierda de la anterior figura servirá para impedir que la corredera pueda desplazarse hacia adelante, además de contar con un sistema similar al del gatillo con el que poder incorporar un muelle. Dado que la base de la pistola en la parte que encaja con la corredera cuenta también con otro saliente, se impedirá a mayores que la corredera rebase la parte posterior del arma cuando se trate de accionar el interruptor de recarga.

Finalmente, con el propósito de verificar si el diseño había sido acertado y el conjunto de componentes podían incorporarse dentro de este, se realizaron los diseños a escala de los componentes. En la **Figura 62** puede observarse el diseño de los componentes realizado mediante la herramienta Tinkercad.



*Figura 61. Diseño tridimensional de componentes.*

De izquierda a derecha se citan los componentes: interruptores de final de carrera, circuitos integrados, motor solenoide, botones, motor vibrador, display LED y conector mini USB a USB 3.0.

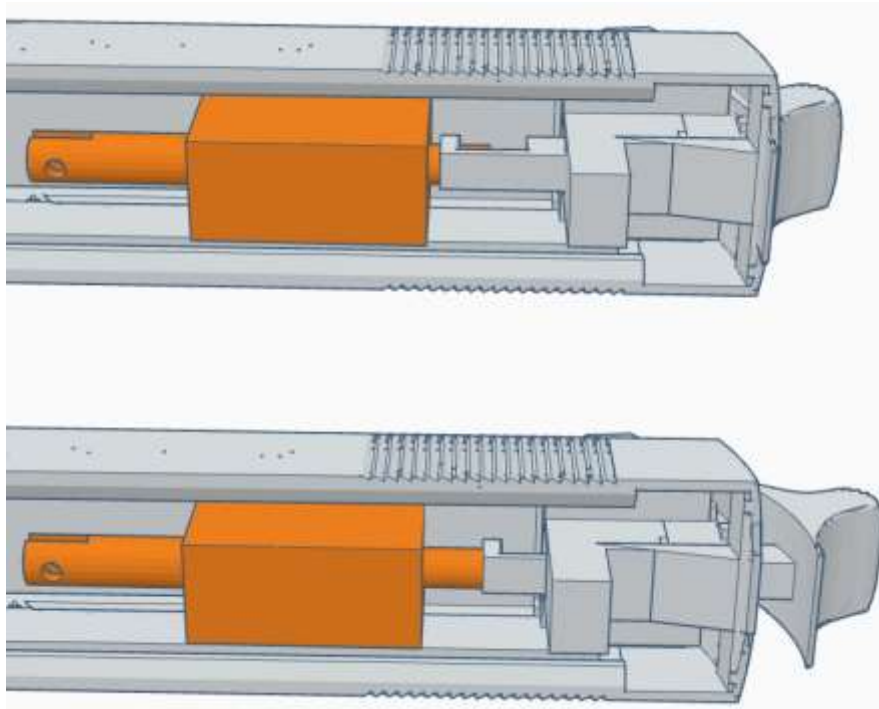
Cabe destacar que se ha prescindido de algunos elementos que cuentan con conexiones en la PCB, siendo estos los diodos LED y el interruptor de modo automático. La decisión de eliminar estos elementos ha venido propiciada principalmente por una cuestión estética. Y es que, sería necesario realizar perforaciones en la corredera para poder incorporar estos elementos. Dichas perforaciones empeorarían de forma sustancial el aspecto final del producto, haciéndolo menos atractivo. A pesar de esto, la PCB cuenta con estas conexiones incorporadas y se podrían añadir en futuras implementaciones de la carcasa.

Continuando con la explicación del diseño, se pasará a explicar una modificación llevada a cabo que dista de las explicaciones aportadas en el resto del proyecto. Esta modificación se basa en la interacción del solenoide con la corredera.

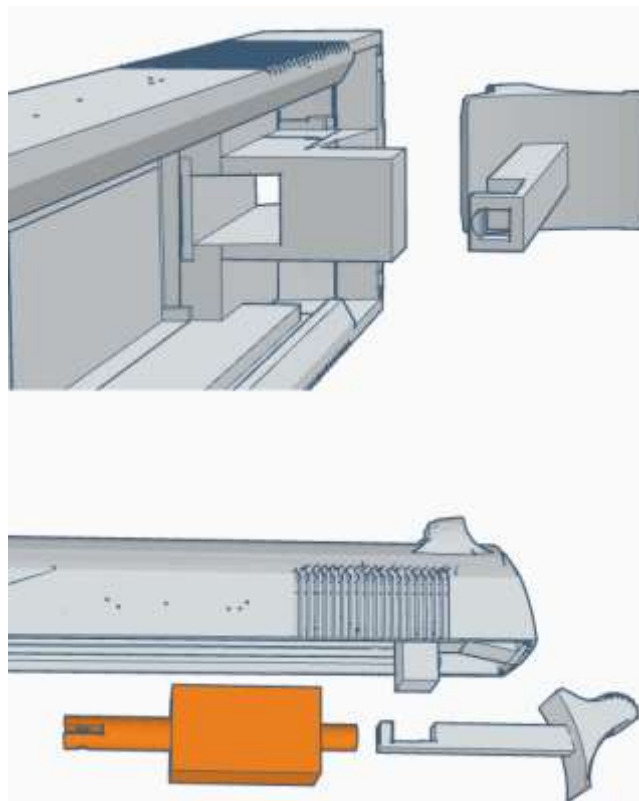
Tras haber llevado a la práctica múltiples realizaciones, se ha verificado que el motor solenoide era capaz de desplazar el peso de la corredera. Por el contrario, a la hora de realizar el montaje completo, el motor era incapaz de mover la corredera debido a las fuerzas originadas por el rozamiento de la corredera en la pieza de la base y el muelle incorporado para el retroceso.

Como consecuencia, se ha recurrido a realizar una perforación en la corredera e incluir una pieza móvil. De este modo, cuando se accione el gatillo, la corredera no se moverá y como contraprestación, el movimiento del solenoide irá dirigido a una pieza móvil en la

que se incorporará el martillo del arma. En las **Figuras 63** y **64** se aporta una explicación gráfica sobre el funcionamiento anteriormente descrito.



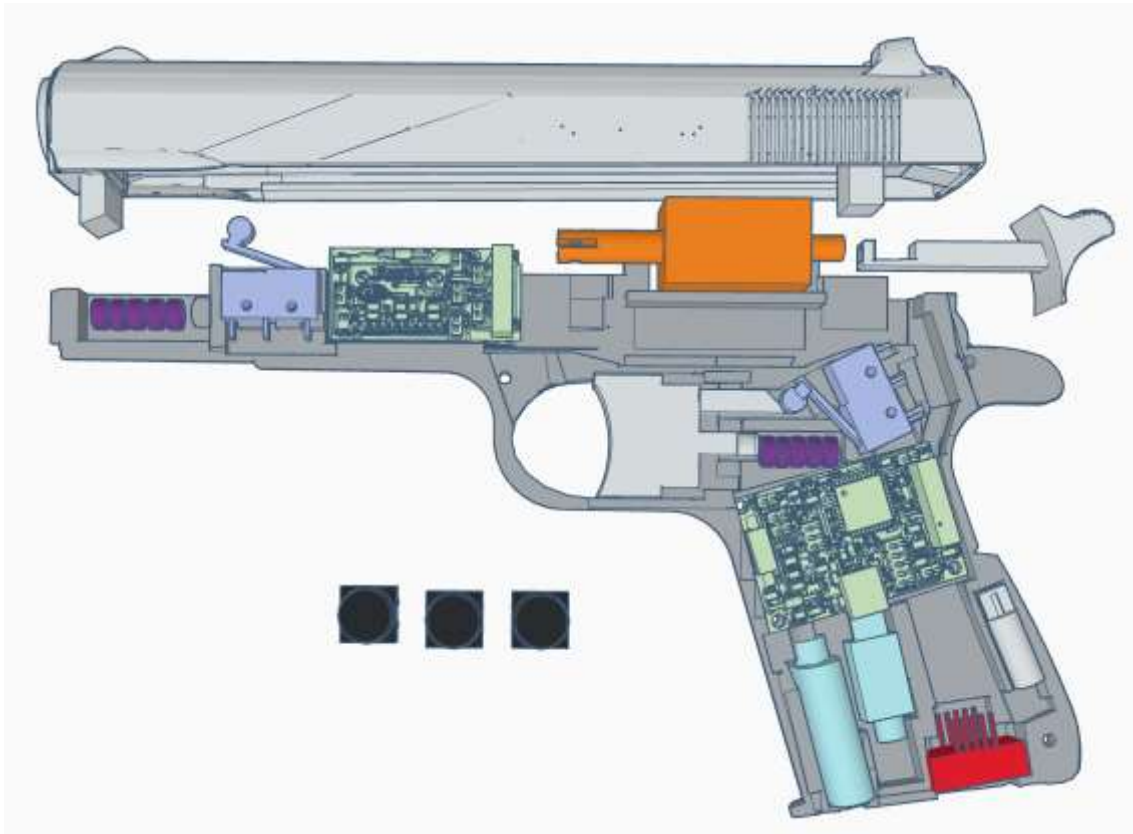
**Figura 62.** Descripción gráfica del accionamiento del motor solenoide.



**Figura 63.** Vista tridimensional sobre la incorporación de la pieza móvil del martillo.



Como conclusión a esta sección del documento, se aporta una vista de la sección del diseño en la que se incorporan todos los componentes. Esta vista puede apreciarse en la **Figura 65**. Los elementos mostrados en color púrpura corresponden a los muelles que permitirán devolver a su posición inicial las piezas móviles.



*Figura 64. Diseño global de la carcasa con sus componentes principales.*

## 5.2 Configuraciones de impresión

Tras haber realizado el diseño tridimensional, surge como condición necesaria aclarar algunas dudas sobre cómo se llevará a cabo el proceso de fabricación, cuáles serán los materiales empleados, cómo se realizarán los soportes para el diseño tridimensional, etc.

Esta sección pretende mostrar la experiencia llevada a cabo en relación a los procesos de impresión tridimensional y tiene como objetivo dar a conocer la dificultad y profundidad de una materia hasta ahora desconocida. Para ello se tratará de proporcionar una perspectiva de futuro que permitirá tener en cuenta diversos factores a la hora de planificar de manera global nuevas iteraciones del proyecto.

Inicialmente, se ha decidido, teniendo en cuenta que existían posibles fallos en el diseño, imprimir parte de la pieza de la base, con el objetivo de familiarizarse con los procesos llevados a cabo, la utilización de impresoras y adquirir conocimientos sobre el tema.

Uno de los problemas principales a la hora de incorporar el diseño tridimensional a una impresora para su creación, era la selección de un soporte o base que permitiese al extrusor de la misma alcanzar las partes que en el diseño se encontraban en estado de “levitación”.

La incorporación de estos, y otros muchos elementos necesarios para la impresión, no se realizan de forma manual. Existen múltiples programas que realizan estas tareas de forma automatizada. Uno de los más empleados en el sector de la impresión tridimensional es el programa Ultimaker Cura [48]. Este programa permite asignar las configuraciones de la impresora 3D que se va a emplear, con el propósito de que la impresión sea lo más fiel posible al diseño asistido por ordenador, ajustándose a las necesidades de uso de la pieza.

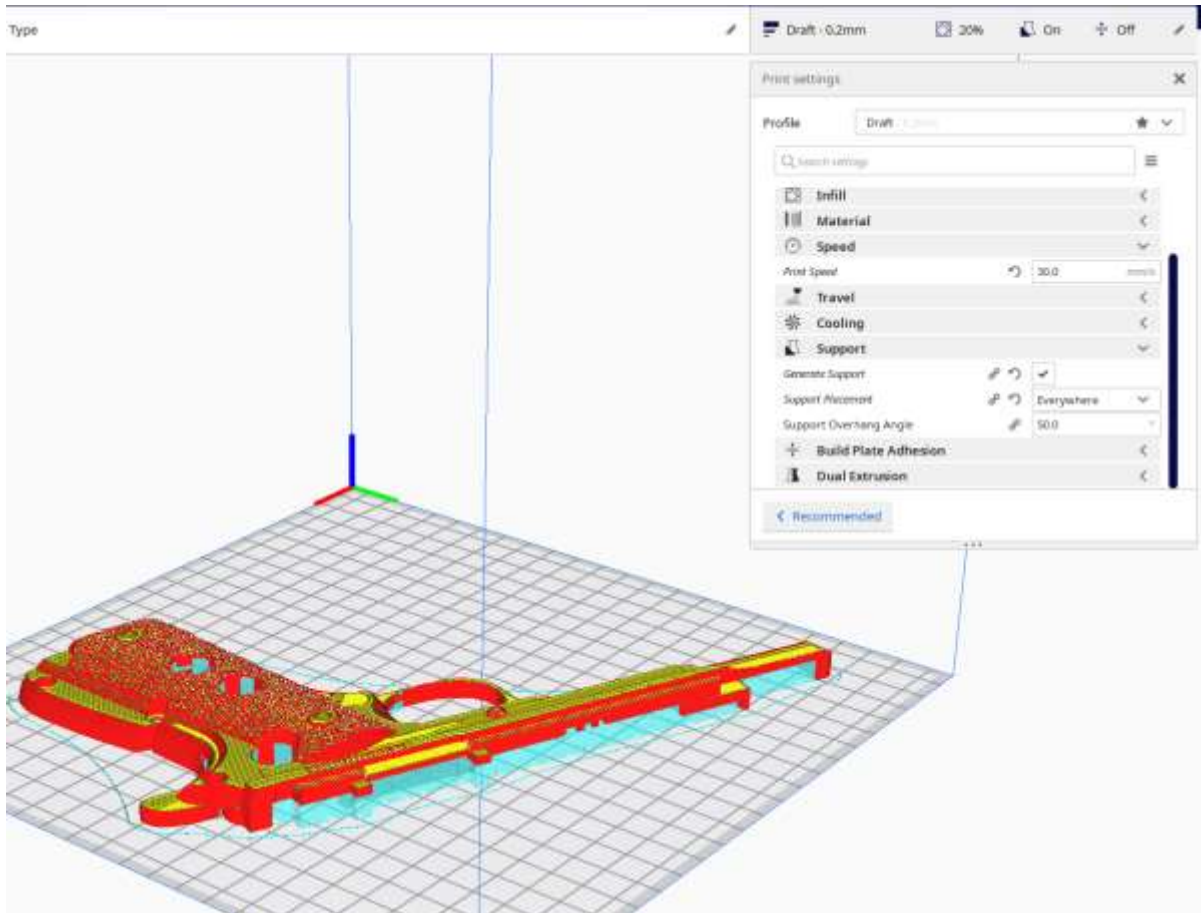
A la hora de trabajar con el programa anteriormente citado, es necesario incorporar el tipo de impresora que se utilizará durante la creación de la pieza. En este caso se ha utilizado un modelo de impresora conocido como **Biqu B1**.

En el programa Ultimaker Cura, uno de los pasos iniciales se fundamentará en configurar el tipo de impresora utilizada en el proceso. Para llevar a cabo la configuración, será necesario indicar el tipo de modelo, tamaño de la cama caliente, tipo de extrusor, diámetro del extrusor y otros muchos parámetros de interés como el diámetro del filamento, o tipo de material.

La impresora empleada posee una cama caliente, la cual será configurada a 60°C, de unas dimensiones de 235x235x270mm. Además, posee un único extrusor con un *nozzle* de 0.4mm de diámetro de salida, en el que se empleará un plástico de ácido poliláctico (PLA) blanco de 1.75mm de diámetro.

Una vez configurada la impresora en el programa Ultimaker Cura, se ha descargado la pieza en formato *STereoLithography* (STL) o “.STL” desde la página web Thinkercad. Tras haber realizado la incorporación tridimensional de la pieza, se ha procedido a generar los soportes y modificar algunos de los parámetros de impresión. Los soportes se han configurado con un ángulo de 50° a fin de facilitar la retirada de estos. Por otro lado, se ha reducido la velocidad de impresión con el objetivo de asegurar que los detalles más pequeños se imprimen de manera correcta.

En la **Figura 66** se muestra un ejemplo de una de las piezas que conforman la carcasa del dispositivo, junto con sus configuraciones.



**Figura 65.** Preparación de impresión tras aplicar el "rebanado" de las distintas capas.

La figura mostrada es el producto de haber aplicado el *slicing* o “rebanado” con las configuraciones elegidas. Este proceso se encarga de generar las distintas capas que imprimirá la impresora para las distintas alturas (eje z). De esta forma se pueden generar los ficheros de código que son comprensibles para la impresora. Estos ficheros poseen la extensión “.gcode” y son ficheros de texto en los que se definen las posiciones en las que deberá colocarse el *nozzle* de la impresora para imprimir, además de especificarse otros parámetros como la velocidad de extrusión, la temperatura de la cama caliente o la versión de *gcode* elegida. En la **Figura 67** puede apreciarse parte del contenido del fichero generado para la pieza de la **Figura 66**.

Una vez generados los ficheros “.gcode”, únicamente quedará incorporarlos a una memoria, en nuestro caso una microSD, para después poder seleccionar la impresión de la pieza desde la impresora.

```
G1 F900 X124.966 Y41.861 E0.02521
G1 X125.259 Y41.449 E0.05043
G1 X125.613 Y41.088 E0.07566
G1 X126.02 Y40.788 E0.10089
G1 X126.345 Y40.611 E0.11935
G1 X126.92 Y40.341 E0.15104
G1 X127.622 Y40.07 E0.18858
G1 X128.366 Y39.865 E0.22708
G1 X130.736 Y39.248 E0.34926
G1 X130.784 Y39.23 E0.35182
G1 X131.485 Y39.053 E0.38789
```

**Figura 66.** Contenido del fichero .gcode generado tras slicing mediante Ultimaker Cura.

A la hora de llevar a cabo una impresión, el tiempo empleado en la configuración puede tener un gran peso. A pesar de esto, se ha puesto en valor que no se necesitan grandes conocimientos ni configuraciones de elevada precisión para conseguir un diseño funcional. Pero sí ha de tenerse en cuenta el factor temporal, dado que se han cargado los diseños en el programa Ultimaker Cura para tratar de hacer una estimación temporal de la impresión. El conjunto completo de piezas alcanzaba entorno a las 37 horas. Un factor muy a tener en cuenta de cara a modificaciones de diseño, y a la gestión temporal del proyecto.

Aunque podríamos realizar cientos de optimizaciones, nos centraremos en la que nos proporcione un diseño que goce de cierto grado de vistosidad, pero que no requiera configuraciones complicadas y por supuesto que sea práctico. Desde un punto de vista de ingeniería se tratará de equilibrar la balanza entre calidad en el acabado, funcionamiento de la carcasa, e inversión temporal. Ya que el objetivo del proyecto no se basa únicamente en la implementación tridimensional de un diseño, sino en la concepción global de un producto electrónico.

### 5.3 Postprocesado

Una vez impresas las piezas en tres dimensiones será necesario realizar algunas modificaciones en estas con la finalidad de mejorar la calidad de la carcasa.

En primera instancia, se hace necesario llevar a cabo la retirada de los soportes de impresión. En la **Figura 68** se puede visualizar una pieza en la que se está llevando a cabo la eliminación de los soportes.



**Figura 67.** Visualización del proceso de retirada de soportes.

Una vez retirados los soportes ya contaremos con las piezas diseñadas. Sin embargo, en algunos casos, la impresora 3D empleada no ha sido tan precisa como el diseño generado mediante herramientas CAD. Es por esto, que se ha decidido aplicar masilla en algunas de las partes de las piezas con el objetivo de suavizar los bordes y aportar mayor calidad al diseño final. En la **Figura 69** puede apreciarse una comparativa entre piezas sin el aplique de masilla y con él.



**Figura 68.** Comparación de piezas tras la aplicación de masilla.

En la figura mostrada anteriormente parece no haber una gran diferencia. Esto es cierto a priori, pero tras realizar un lijado de las piezas las diferencias son notables en la textura.

Como último paso en el postprocesado de las piezas se ha aplicado pintura negra, asemejando de esta forma la carcasa plástica a una pistola real.

## 6. COSTES HUMANOS

A la hora de entrar a valorar los costes humanos se ha tenido en cuenta una planificación temporal inicial. Esta planificación ha sido modificada a lo largo del desarrollo del proyecto. Para ello se ha empleado el programa de planificación de proyectos GanttProject [49].

Mediante este programa, se han establecido, inicialmente, unos márgenes temporales para cada uno de los apartados del proyecto. Posteriormente, se han ido modificando en función del trabajo práctico correspondiente. A continuación, se mostrarán los tiempos empleados en los distintos apartados del proyecto. Estos tiempos nos servirán como nueva referencia a la hora de planificar nuevas iteraciones del proyecto o proyectos similares. Las **Figuras 70, 71 y 72** muestran los tiempos empleados durante la realización de las distintas tareas del proyecto.

Se puede observar cómo algunos de los tiempos aparecen marcados en color verde, debido a que se han podido realizar en el tiempo inicialmente estimado o han supuesto un tiempo inferior al previsto. Por otro lado, las barras de color rojo representan retrasos y ampliaciones de desarrollo en dichas tareas.

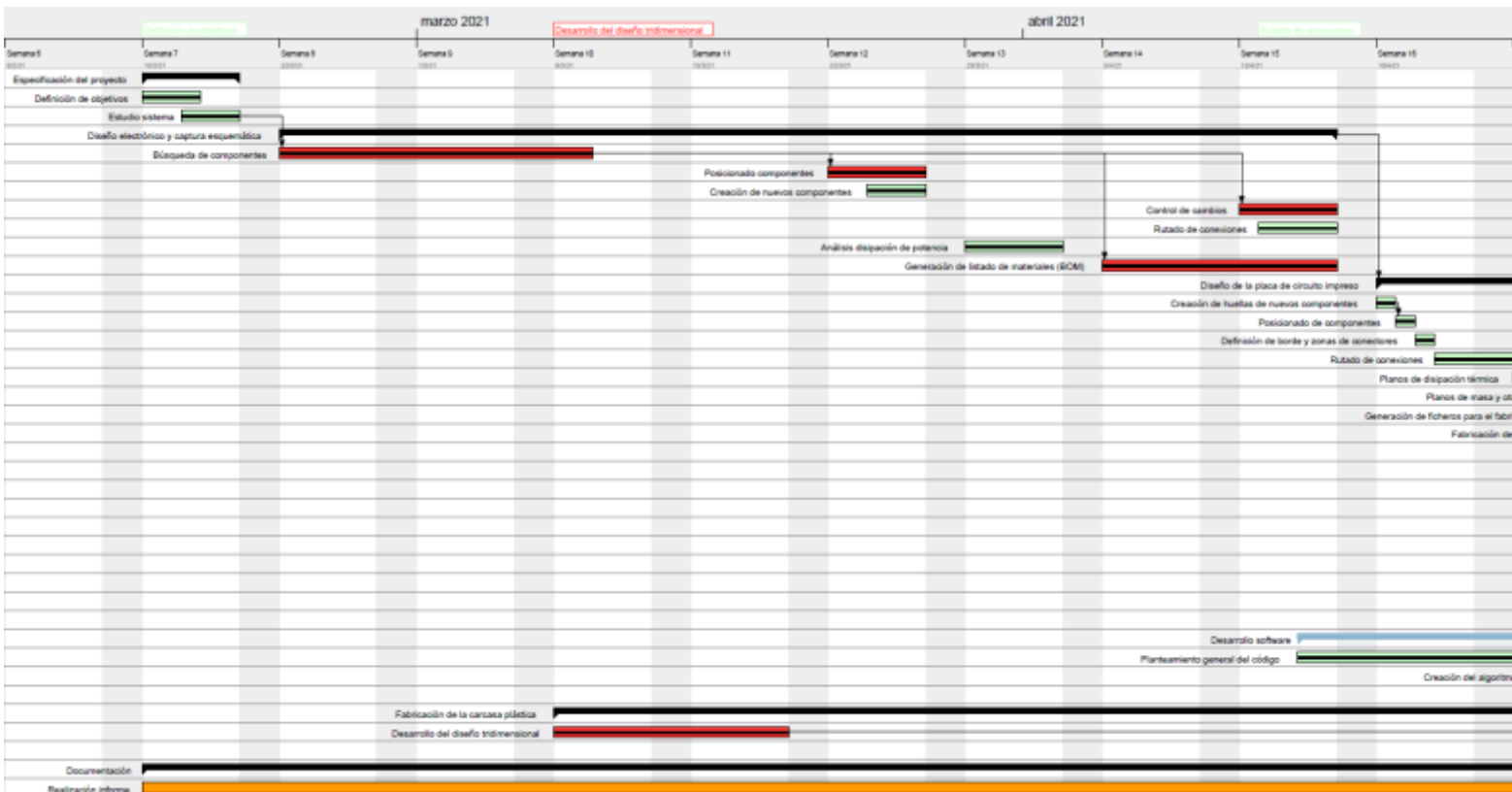


Figura 69. Representación del coste temporal de trabajo - Parte 1.

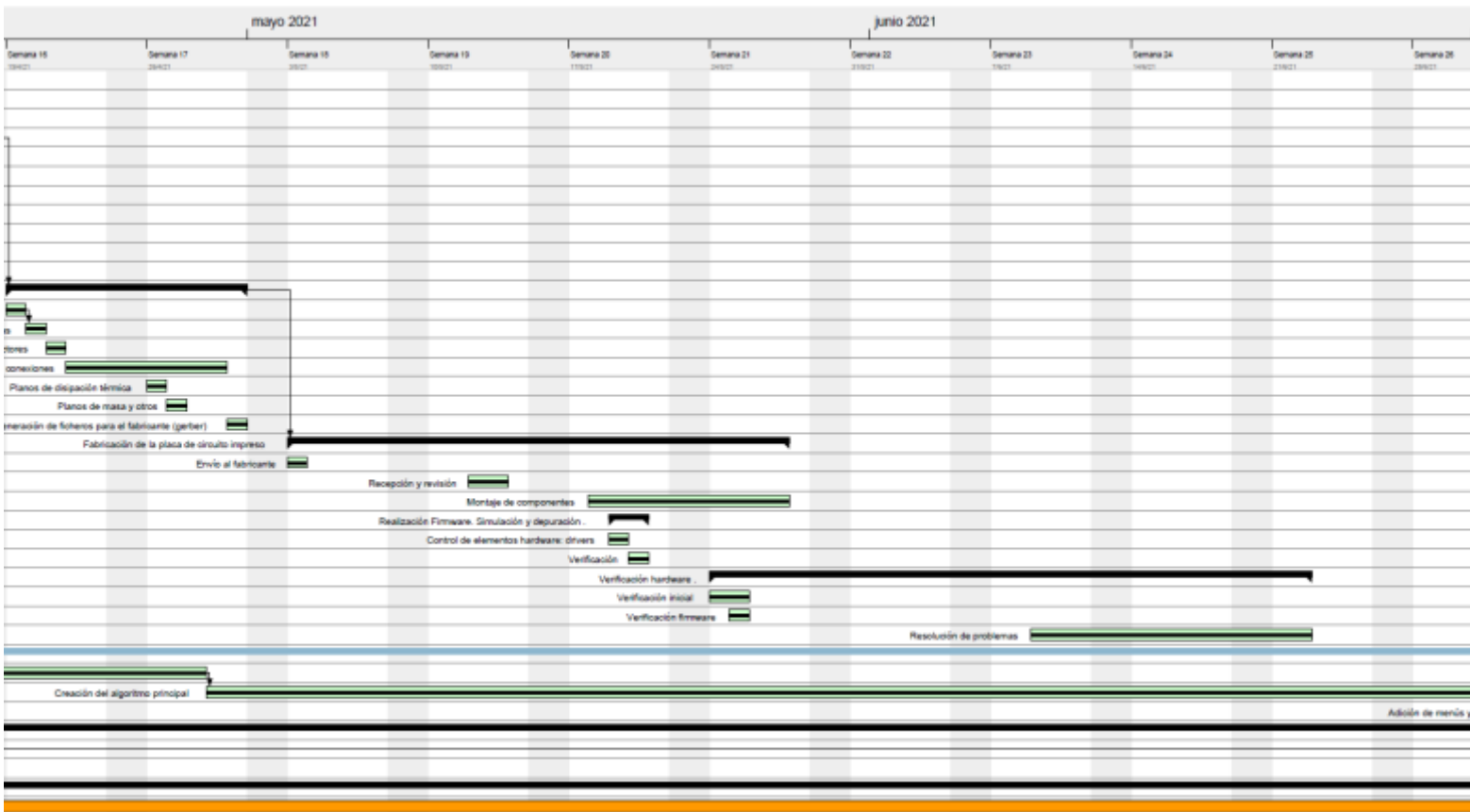


Figura 71. Representación del coste temporal de trabajo - Parte 2.

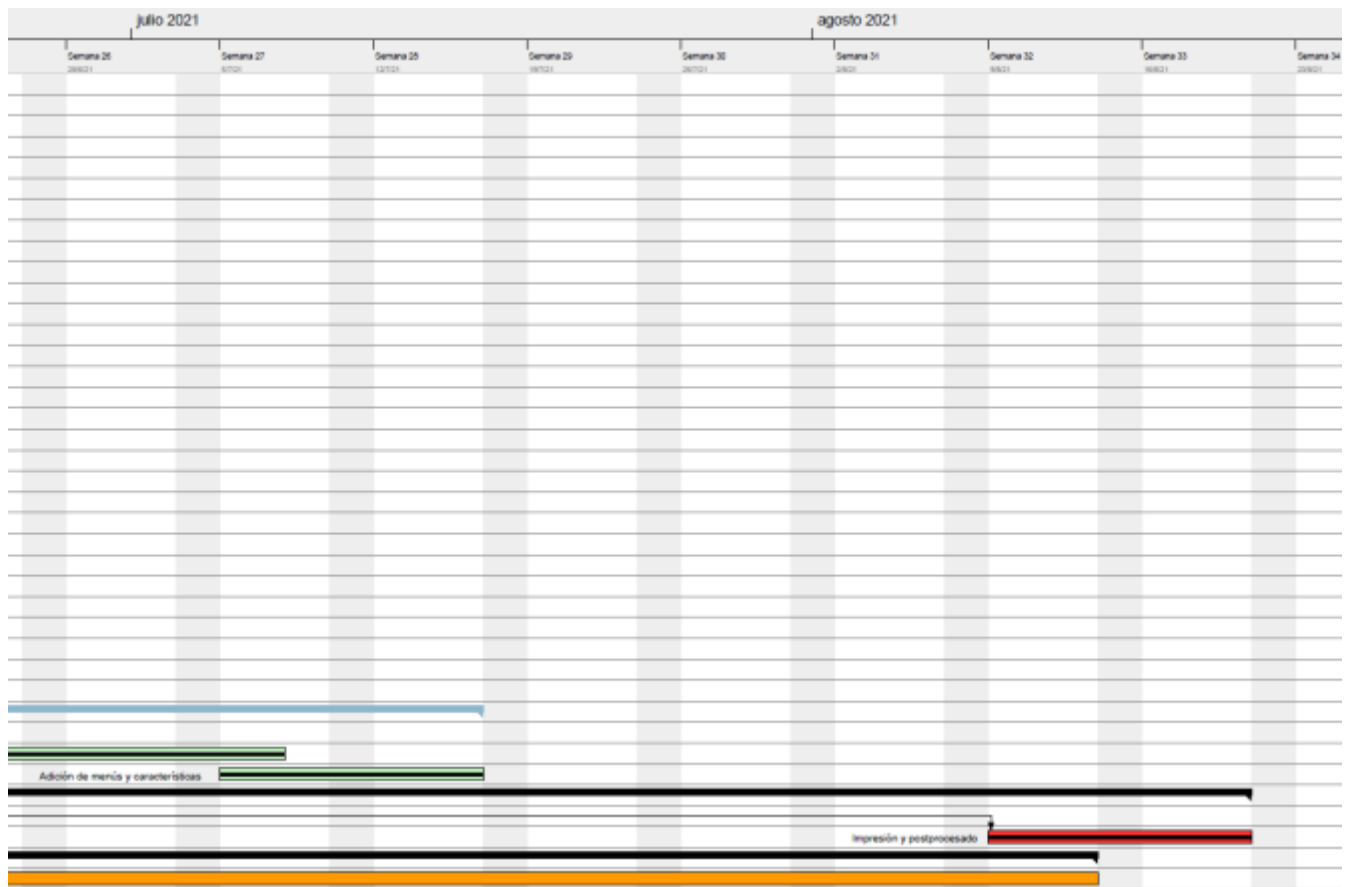


Figura 70. Representación del coste temporal de trabajo - Parte 3.

Finalmente, a fin de facilitar la visualización de los apartados y los tiempos empleados, se aporta un listado en forma textual en la **Figura 73**.

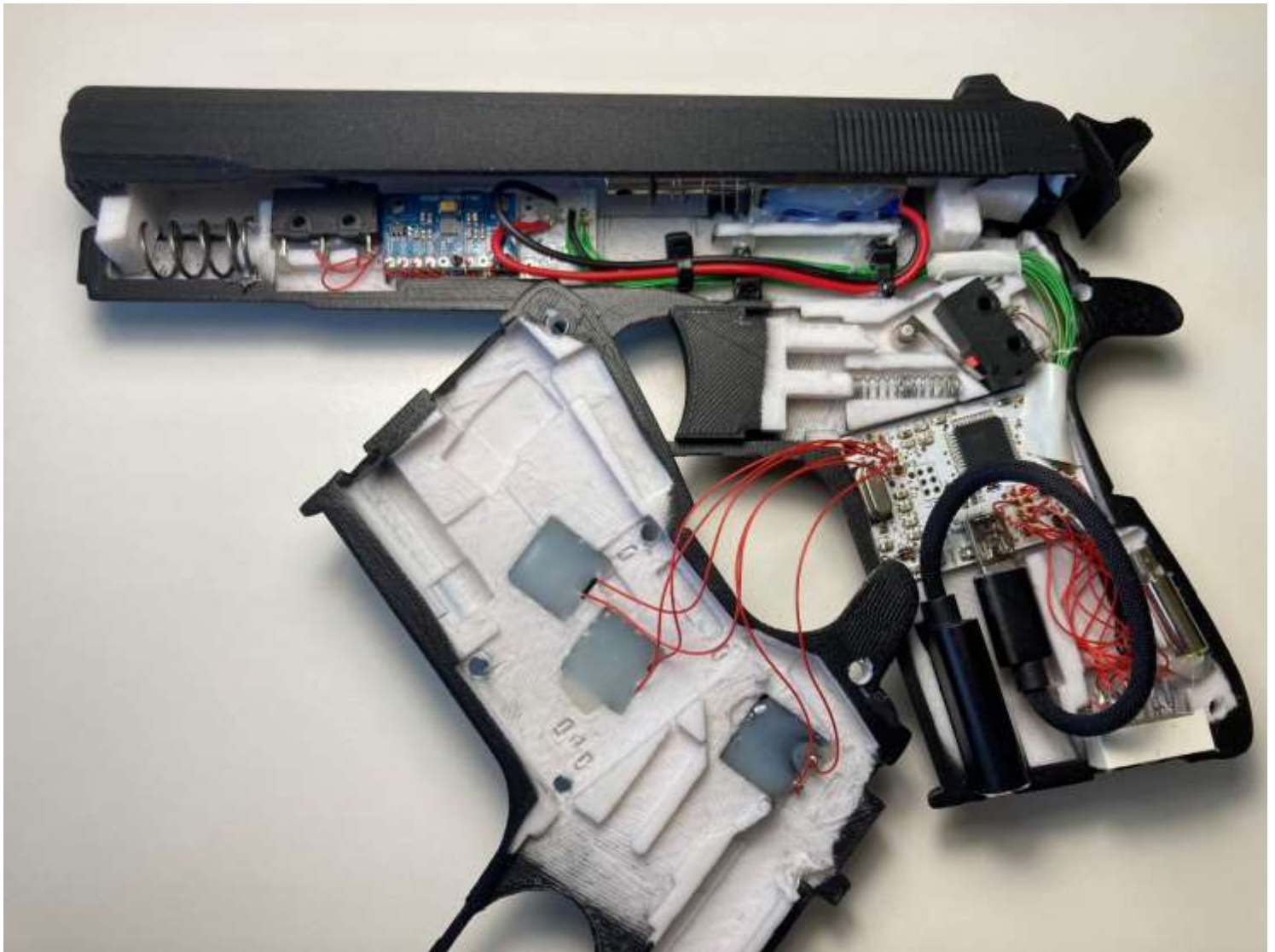
<b>NOMBRE</b>	<b>INICIO</b>	<b>FIN</b>
<b>Especificación del proyecto</b>	15/2/21	19/2/21
Definición de objetivos	15/2/21	17/2/21
Estudio sistema	17/2/21	19/2/21
<b>Diseño electrónico y captura esquemática</b>	22/2/21	16/4/21
Búsqueda de componentes	22/2/21	9/3/21
Posicionado de componentes	22/3/21	26/3/21
Creación de nuevos componentes	24/3/21	26/3/21
Control de cambios	12/4/21	16/4/21
Rutado de conexiones	13/4/21	16/4/21
Análisis disipación de potencia	29/3/21	2/4/21
Generación de listado de materiales (BOM)	5/4/21	16/4/21
<b>Diseño de la placa de circuito impreso</b>	19/4/21	30/4/21
Creación de huellas de nuevos componentes	19/4/21	19/4/21
Posicionado de componentes	20/4/21	20/4/21
Definición de borde y zonas de conectores	21/4/21	21/4/21
Rutado de conexiones	22/4/21	29/4/21
Planos de disipación térmica	26/4/21	26/4/21
Planos de masa y otros	27/4/21	27/4/21
Generación de ficheros para el fabricante (gerber)	30/4/21	30/4/21
<b>Fabricación de la placa de circuito impreso</b>	3/5/21	27/5/21
Envío al fabricante	3/5/21	3/5/21
Recepción y revisión	12/5/21	13/5/21
Montaje de componentes	18/5/21	27/5/21
<b>Realización Firmware. Simulación y depuración.</b>	19/5/21	20/5/21
Control de elementos hardware: drivers	19/5/21	19/5/21
Verificación	20/5/21	20/5/21
<b>Verificación hardware</b>	24/5/21	22/6/21
Verificación inicial	24/5/21	25/5/21
Verificación firmware	25/5/21	25/5/21
Resolución de problemas	9/6/21	22/6/21
<b>Desarrollo software</b>	15/4/21	16/7/21
Planteamiento general del código	15/4/21	28/4/21
Creación del algoritmo principal	29/4/21	7/7/21
Adición de menús y características	5/7/21	16/7/21
<b>Fabricación de la carcasa plástica</b>	8/3/21	20/8/21
Desarrollo del diseño tridimensional	8/3/21	19/3/21
Impresión y postprocesado	9/8/21	20/8/21
<b>Documentación</b>	15/2/21	13/8/21
Realización informe	15/2/21	13/8/21

**Figura 73.** Listado generado con GanttProject sobre el flujo temporal del proyecto.



## 7. IMPLEMENTACIÓN FINAL

Una vez realizado el montaje de componentes y la impresión de la carcasa, se ha procedido con la incorporación del circuito en el interior de esta. En las siguientes figuras se refleja el resultado físico final del producto, así como la incorporación de la circuitería en el mismo.



*Figura 74. Sección del producto final tras incorporar el circuito electrónico a la carcasa.*



*Figura 75. Producto final tras la incorporación del circuito y su ensamblado.*

## **SIGLAS.**

*Computer Assisted Design (CAD)*

*Operating System (OS)*

*Human Interface Devices (HID)*

*Universal Serial Bus (USB)*

*Light-Emitting Diode (LED)*

*Single Pole Double Throw (SPTD)*

*Multi-Chip Module (MCM)*

*Quad-Flat No-Leads (QFN)*

*Microelectromechanical Systems (MEMS)*

*Analog to Digital Converter (ADC)*

*Inter-Integrated Circuit (I<sup>2</sup>C)*

*Serial Peripheral Interface (SPI)*

*Frames Per Second (FPS)*

*Dot Per Inch (DPI)*

*Printed Circuit Board (PCB)*

*Pulse-Width Modulation (PWM)*

*In-System Programming (ISP)*

*Serial Clock (SCLK)*

*Master Output Slave Input (MOSI)*

*Master Input Slave Output (MISO)*

*Select (SS)*

*Serial Data (SDA)*

*Two Wire Interface (TWI)*

*Integrated Development Environment (IDE)*

*Direct Current (DC)*

*Resistance-Capacitor (RC)*

*Thin Quad Flat Package (TQFP)*

*Electrical Computer Aided Design (ECAD)*

*Entrada/Salida (E/S)*

*Signal to Noise Ratio (SNR)*

*Equivalent Series Resistance (ESR)*

*In-Circuit Serial Programming (ICSP)*

*ElectroMagnetic Interference (EMI)*

*Digital Motion Processor (DMP)*

*Bill Of Materials (BOM)*

*Inertial Measurement Unit (IMU)*

*WatchDog Timer (WDT)*

*High Definition (HD)*

*Full HD (FHD)*

*Quad HD (QHD)*

*UltraWide (UW)*

*Ultra HD (UHD)*

*Ácido poliláctico (PLA)*

*STereoLithography (STL)*

## REFERENCIAS.

- [1] “Human Interface Devices (HID) Information | USB-IF.”  
<https://www.usb.org/hid> (accessed Jun. 23, 2021).
- [2] Mike Bergman *et al.*, “Device Class Definition for Human Interface Devices (HID), Firmware Specification Version 1.11,” 2001. Accessed: Jun. 23, 2021. [Online]. Available:  
[https://www.usb.org/sites/default/files/documents/hid1\\_11.pdf](https://www.usb.org/sites/default/files/documents/hid1_11.pdf).
- [3] “About USB-IF | USB-IF.” <https://www.usb.org/about> (accessed Jun. 23, 2021).
- [4] J. Sebastián M Sebastián, M, “Las Leyes de Newton de la mecánica: Una revisión histórica y sus implicaciones en los textos de enseñanza,” 2013, Accessed: Jun. 24, 2021. [Online]. Available: <https://core.ac.uk/download/pdf/71022947.pdf>.
- [5] “snap switches DG sub-sub- miniature switch.” .
- [6] “GTIWUNG Interruptor de Rodillo .” [https://www.amazon.es/GTIWUNG-Interruptor-enrutador-Impresora-125V-250/dp/B08LD68FYQ/ref=sr\\_1\\_6?\\_\\_mk\\_es\\_ES=ÅMÅŽÕÑ&dchild=1&keywords=interruptor+final+carrera&qid=1616608477&sr=8-6](https://www.amazon.es/GTIWUNG-Interruptor-enrutador-Impresora-125V-250/dp/B08LD68FYQ/ref=sr_1_6?__mk_es_ES=ÅMÅŽÕÑ&dchild=1&keywords=interruptor+final+carrera&qid=1616608477&sr=8-6) (accessed Jul. 06, 2021).
- [7] “Data Sheet for 5mm Super Bright Amber Yellow LED 5C3 Series Class : P Part No : RF- WW05C3AYP4-N2 Data Sheet For 5mm Super Bright Amber Yellow LED – Class : P.” .
- [8] “5mm Straw Hat LED.”  
<https://es.aliexpress.com/item/4000896062123.html?spm=a2g0s.8937460.0.0.336c2e0eftuvwS> (accessed Jul. 06, 2021).
- [9] L. SHENZHEN ZONHEN ELECTRIC APPLIANCES Co., “SPECIFICATION PART NO.: ZH0-0420S-05A4.5.” .
- [10] “solenoid electromagnet HS-0730B DC6V 5N 10mm stroke push pull frame – Heschen.” <https://heschen.com/es/products/hs-0730b-dc-6v> (accessed Jul. 06, 2021).
- [11] L. ZHEJIANG YUESUI ELECTRON STOCK CO., “Coin Type Vibration Motor B1034.FL45-00-015.” 2016, [Online]. Available:  
<https://cdn.sparkfun.com/datasheets/Robotics/B1034.FL45-00-015.pdf>.

- [12] “Micro Motor Sin Núcleo NdFeB 615.”  
[https://es.aliexpress.com/item/1005001297087992.html?spm=a2g0o.detail.1000013.3.5106b462hjFgRj&gps-id=pcDetailBottomMoreThisSeller&scm=1007.13339.169870.0&scm\\_id=1007.13339.169870.0&scm-url=1007.13339.169870.0&pvid=f98db6b5-2355-4426-9322-ba26597dcd88&\\_t=gps-id:pcDetailBottomMoreThisSeller,scm-url:1007.13339.169870.0,pvid:f98db6b5-2355-4426-9322-ba26597dcd88,ttp\\_buckets:668%20%23131923%2339\\_668%20%23131923%2339\\_668%23888%233325%2312\\_668%23888%233325%2312\\_668%232846%238108%231977\\_668%235811%2327174%2320\\_668%232717%237563%23581\\_668%231000022185%231000066059%230\\_668%233468%2315617%23825\\_668%232846%238108%231977\\_668%235811%2327174%2320\\_668%232717%237563%23581\\_668%233164%239976%23811\\_668%233468%2315617%23825](https://es.aliexpress.com/item/1005001297087992.html?spm=a2g0o.detail.1000013.3.5106b462hjFgRj&gps-id=pcDetailBottomMoreThisSeller&scm=1007.13339.169870.0&scm_id=1007.13339.169870.0&scm-url=1007.13339.169870.0&pvid=f98db6b5-2355-4426-9322-ba26597dcd88&_t=gps-id:pcDetailBottomMoreThisSeller,scm-url:1007.13339.169870.0,pvid:f98db6b5-2355-4426-9322-ba26597dcd88,ttp_buckets:668%20%23131923%2339_668%20%23131923%2339_668%23888%233325%2312_668%23888%233325%2312_668%232846%238108%231977_668%235811%2327174%2320_668%232717%237563%23581_668%231000022185%231000066059%230_668%233468%2315617%23825_668%232846%238108%231977_668%235811%2327174%2320_668%232717%237563%23581_668%233164%239976%23811_668%233468%2315617%23825) (accessed Jul. 06, 2021).
- [13] InvenSense Inc., “MPU-9250 Product Specification Revision 1.1.” 1745 Technology Drive, San Jose, CA 95110 U.S.A., 2016, [Online]. Available: [www.invensense.com](http://www.invensense.com).
- [14] “MPU 9250 9 axis.”  
<https://es.aliexpress.com/item/4000093478742.html?spm=a2g0o.cart.0.0.55993c005pIlfx&mp=1> (accessed Jul. 06, 2021).
- [15] ATMEL, “ATmega16U4 / ATmega32U4.” 2016, [Online]. Available: [https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7766-8-bit-AVR-ATmega16U4-32U4\\_Datasheet.pdf](https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7766-8-bit-AVR-ATmega16U4-32U4_Datasheet.pdf).
- [16] Compaq *et al.*, “Universal Serial Bus Specification,” *Group*, 2000, [Online]. Available: <https://www.usb.org/document-library/usb-20-specification%0A%0A>.
- [17] Microchip, “What is TWI? How to Configure the TWI for I2C Communication,” *Microchip Technol. Inc.*, 2018, [Online]. Available: <http://ww1.microchip.com/downloads/en/DeviceDoc/90003181A.pdf>.
- [18] Arduino, “What is Arduino? | Arduino.”  
<https://www.arduino.cc/en/Guide/Introduction> (accessed Jul. 01, 2021).
- [19] “ATMEGA32U4-AUR .”  
<https://es.aliexpress.com/item/32916694199.html?spm=a2g0s.8937460.0.0.2bbd2>

- e0esBeger (accessed Jul. 06, 2021).
- [20] LabCenter, “PCB Design and Circuit Simulator Software - Proteus.”  
<https://www.labcenter.com/> (accessed Jul. 08, 2021).
- [21] SamacSys Ltd, “SamacSys - Electronic Component CAD Models – Symbols | Footprints | 3D Models.” <https://www.samacsys.com/> (accessed Jul. 08, 2021).
- [22] SamacSys Ltd, “Proteus - SamacSys - Configuring Library Loader for Proteus 8.8 or later.” <https://www.samacsys.com/proteus/> (accessed Jul. 08, 2021).
- [23] SamacSys Ltd, “Component Search Engine: Free Symbols, footprints, & 3D models.” <https://componentsearchengine.com/> (accessed Jul. 08, 2021).
- [24] Atmel, “AVR042 : AVR hardware design considerations,” *Appl. Note*, 2015, [Online]. Available: [http://ww1.microchip.com/downloads/en/appnotes/atmel-2521-avr-hardware-design-considerations\\_applicationnote\\_avr042.pdf](http://ww1.microchip.com/downloads/en/appnotes/atmel-2521-avr-hardware-design-considerations_applicationnote_avr042.pdf).
- [25] M. A. Cebrián, E. T. S. I. de Telecomunicación, and U. de Valladolid, “Fundamentos de Electrónica Tema 2: el diodo,” 2017.
- [26] ONSemiconductor, “1N91x, 1N4x48, FDLL914, FDLL4x48 Small Signal Diode,” 2002, [Online]. Available: [www.onsemi.com](http://www.onsemi.com).
- [27] M. A. Cebrián, E. T. S. I. de Telecomunicación, and U. de Valladolid, “Fundamentos de Electrónica Tema 4: Transistores de Efecto de Campo,” 2017.
- [28] V. L. P. Semiconductor, “2N7002 N-Channel Enhancement Mode Field Effect Transistor.pdf.” .
- [29] ONSemiconductor, “NDS331N N-Channel Logic Level Enhancement Mode Field Effect Transistor,” 2017.
- [30] MICREL, “MIC5219 500mA-Peak Output LDO Regulator,” p. 283, 1386, [Online]. Available: [www.micrel.com](http://www.micrel.com).
- [31] “Pantalla LED de 1/2/3/4/5 Bit .”  
<https://es.aliexpress.com/item/4000939769059.html?spm=a2g0s.8937460.0.0.15de2e0eUS7rLY> (accessed Jul. 09, 2021).
- [32] Arduino, “Arduino as ISP and Arduino Bootloaders | Arduino.”  
<https://www.arduino.cc/en/Tutorial/BuiltInExamples/ArduinoISP> (accessed Jul. 09, 2021).

- [33] U. S. B. Power and D. Specification, “Universal Serial Bus Power Delivery Specification Revision: 3.1 Version :1.1 Release date : Juñy 2021,” 2021, [Online]. Available: <https://www.usb.org/document-library/usb-power-delivery>.
- [34] PCBWay, “Fabricación de PCB - Servicio completo para prototipos de PCB personalizados.” <https://www.pcbway.es/> (accessed Jul. 19, 2021).
- [35] Arduino, “Mouse - Arduino Reference.” <https://www.arduino.cc/reference/en/language/functions/usb/mouse/> (accessed Aug. 16, 2021).
- [36] jonathanedgcombe (GitHub user), “absmouse/AbsMouse.cpp at master · jonathanedgcombe/absmouse · GitHub.” <https://github.com/jonathanedgcombe/absmouse/blob/master/src/AbsMouse.cpp> (accessed Aug. 16, 2021).
- [37] Arduino, “Keyboard - Arduino Reference.” <https://www.arduino.cc/reference/en/language/functions/usb/keyboard/> (accessed Aug. 16, 2021).
- [38] bolderflight (GitHub user), “GitHub - bolderflight/mpu9250: MPU-9250 sensor driver.” <https://github.com/bolderflight/MPU9250> (accessed Aug. 16, 2021).
- [39] bolderflight (GitHub user), “GitHub - bolderflight/imu: Defines a common interface for BFS IMU sensors.” <https://github.com/bolderflight/imu> (accessed Aug. 16, 2021).
- [40] sigvaldm (GitHub user), “GitHub - sigvaldm/SevenSeg: Arduino 7-segment display library.” <https://github.com/sigvaldm/SevenSeg> (accessed Aug. 16, 2021).
- [41] S. Marholm, “Arduino SevenSeg 1.2.1,” pp. 1–21, 2017, [Online]. Available: <https://raw.githubusercontent.com/sigvaldm/SevenSeg/master/extras/SevenSeg.pdf>.
- [42] Giuseppe Masino - qub1750ul (GitHub user), “GitHub - qub1750ul/Arduino\_SoftwareReset: Arduino library that allows to reset an arduino board from the sketch.” [https://github.com/qub1750ul/Arduino\\_SoftwareReset](https://github.com/qub1750ul/Arduino_SoftwareReset) (accessed Aug. 16, 2021).
- [43] “History of Colt 45 - HISTORY.” <https://www.history.com/topics/industrial->


















- revolution/history-of-colt-45-video (accessed Aug. 18, 2021).
- [44] “M1911 pistol series - Internet Movie Firearms Database - Guns in Movies, TV and Video Games.” [http://www.imfdb.org/wiki/M1911\\_pistol\\_series#Film](http://www.imfdb.org/wiki/M1911_pistol_series#Film) (accessed Aug. 18, 2021).
- [45] “Colt M1911 - Comparison in 60 Different Games - YouTube.” <https://www.youtube.com/watch?v=WV8mAIsg1wE> (accessed Aug. 18, 2021).
- [46] Martin Smith - gustofusion (Thingiverse user), “Colt 1911 Replica Prop - Thingiverse.” <https://www.thingiverse.com/thing:3535584> (accessed Aug. 19, 2021).
- [47] Autodesk, “Tinkercad | Crea diseños digitales 3D con CAD en línea.” <https://www.tinkercad.com/> (accessed Aug. 19, 2021).
- [48] Ultimaker, “Ultimaker Cura: software de impresión 3D potente y fácil de usar .” <https://ultimaker.com/es/software/ultimaker-cura> (accessed Aug. 19, 2021).
- [49] GanttProject, “GanttProject - Free Project Management Application.” <https://www.ganttproject.biz/> (accessed Aug. 21, 2021).

# ANEXO


## 1. Obtención de materiales

A continuación, se muestran las imágenes de compra sobre los elementos adquiridos para el desarrollo del proyecto:


N.º pedido: 3011849915691895 <a href="#">Ver detalles</a> Hora y fecha de pedido: 03:54 Jun. 16 2021		Tienda: LUDAMOS Oficial Store <a href="#">Ver tienda</a>   <a href="#">Contactar vendedor</a>		Importe total: <b>€ 6,44</b>
 200 piezas, Canal N NTR4503N NTR4501N NTR3C21NZ NDS5331N MMBF0201NL MGSF1N02L MGSF2N02EL STR2N2VH5 SOT23 de alta calidad Mosfets [Resumen del pedido] € 4,87 X1 Propiedades de artículo: NDS5331N 20V  	Confirmación recibida	Finalizado	<a href="#">Seguir pedido</a> <a href="#">Añadir a la cesta</a>	
N.º pedido: 8131870450161895 <a href="#">Ver detalles</a> Hora y fecha de pedido: 09:50 May. 11 2021		Tienda: xin cheng electronic <a href="#">Ver tienda</a>   <a href="#">Contactar vendedor</a>		Importe total: <b>€ 0,94</b>
 Mini conector USB SMD de 5 pines hembra, 10 Uds. [Resumen del pedido] € 0,47 X1  	Confirmación recibida	Finalizado	<a href="#">Añadir a la cesta</a>	
N.º pedido: 8130168428021895 <a href="#">Ver detalles</a> Hora y fecha de pedido: 17:03 Apr. 23 2021		Tienda: HocXYZ Oficial Store <a href="#">Ver tienda</a>   <a href="#">Contactar vendedor</a>		Importe total: <b>€ 4,50</b>
 RESORTE DE COMPRESIÓN de acero inoxidable 304 SUS, resorte de retorno de Rotor tipo Y de 1,0mm de diámetro, 10 Uds. [Resumen del pedido] € 3,15 X1 Propiedades de artículo: 1.0mm + 100mm + 8mm  	Confirmación recibida	Finalizado	<a href="#">Añadir a la cesta</a>	
N.º pedido: 3010957604121895 <a href="#">Ver detalles</a> Hora y fecha de pedido: 11:26 Apr. 21 2021		Tienda: EYEWINK 001 Store <a href="#">Ver tienda</a>   <a href="#">Contactar vendedor</a>		Importe total: <b>€ 6,14</b>
 Multímetro AT34 AT35 USB 3.0, voltímetro LCD a color, amperímetro, medidor de corriente de voltaje, batería de carga, probador USB [Resumen del pedido] € 4,80 X1 Propiedades de artículo: AT34  	Confirmación recibida	Finalizado	<a href="#">Añadir a la cesta</a>	
N.º pedido: 3011979091221895 <a href="#">Ver detalles</a> Hora y fecha de pedido: 11:01 Jun. 28 2021		Tienda: BIQU 3DPrinter Global Store <a href="#">Ver tienda</a>   <a href="#">Contactar vendedor</a>		Importe total: <b>€ 228,42</b>
 BIQU-impresora 3D B1, pantalla táctil de 3,5 pulgadas, placa base de 32 bits, hoja de alta precisión, Impresión de fallo de energía [Resumen del pedido] € 228,00 X1 Propiedades de artículo: Black + Czech Republic  	Confirmación recibida	Finalizado	<a href="#">Seguir pedido</a> <a href="#">Añadir a la cesta</a>	

N.º pedido: 3010957604301895 [Ver detalles](#)      Tienda: NIROLLE Official Store      Importe total:   
 Hora y fecha de pedido: 11:26 Apr. 21 2021      [Ver tienda](#) | [Contactar vendedor](#)      **€ 17,33**

 <p>Cable de extensión USB 3,0, Cable de sincronización de datos macho a hembra, Cable de extensión USB, 1m, 2m, 3m, cable de ordenador        [Resumen del pedido]        € 3,99 X2        Propiedades de artículo: USB3.0 PVC Black + 6ft 1.8m   </p>	<p>Confirmación recibida</p> <p>Finalizado</p>	<p><b>Añadir a la cesta</b></p>
 <p>QGeem cable USB 3,0 supervelocidad USB 3,0 de 2,0 macho a macho Cable de extensión USB para radiador de disco duro Cable de datos USB 3,0 Extender        [Resumen del pedido]        € 3,98 X2        Propiedades de artículo: U 3 Cable + 1.8m   </p>	<p>Confirmación recibida</p>	




N.º pedido: 3010957604331895 [Ver detalles](#)      Tienda: Robotsky Official Store      Importe total:   
 Hora y fecha de pedido: 11:26 Apr. 21 2021      [Ver tienda](#) | [Contactar vendedor](#)      **€ 3,57**

 <p>Cable de 5 pines B macho a USB 3,0 hembra adaptador de Cable de Metal Cable de extensión Micro B-USB para coche MP4 Mini USB3.0 a Micro usb-b OTG        [Resumen del pedido]        € 1,22 X2        Propiedades de artículo: Black   </p>	<p>Confirmación recibida</p> <p>Finalizado</p>	<p><b>Añadir a la cesta</b></p>
--	--	---------------------------------




N.º pedido: 3010957604201895 [Ver detalles](#)      Tienda: Shenzhen Quanxingwei Store      Importe total:   
 Hora y fecha de pedido: 11:26 Apr. 21 2021      [Ver tienda](#) | [Contactar vendedor](#)      **€ 3,76**


 <p>10 Uds 40 Pin 1x40 sola fila macho y hembra 2,54 conector de pines divisible de para Arduino negro        [Resumen del pedido]        € 1,09 X2        Propiedades de artículo: Male   </p>	<p>Confirmación recibida</p> <p>Finalizado</p>	<p><b>Añadir a la cesta</b></p>
--	--	---------------------------------

N.º pedido: 3011114479471895 [Ver detalles](#)      Tienda: YT Electronics components co.,Ltd      Importe total:   
 Hora y fecha de pedido: 10:46 Apr. 21 2021      [Ver tienda](#) | [Contactar vendedor](#)      **€ 1,48**

 <p>100 unidades/lote 2N7002LT 2N7002 7002 N-MOSFET canal N-CH SOT23 mejor calidad        [Resumen del pedido]        € 0,67 X1   </p>	<p>Confirmación recibida</p> <p>Finalizado</p>	<p><b>Añadir a la cesta</b></p>
---	--	---------------------------------

N.º pedido: 3011114479321895 [Ver detalles](#)      Tienda: DG-HKDZ-Hardware Store      Importe total:   
 Hora y fecha de pedido: 10:46 Apr. 21 2021      [Ver tienda](#) | [Contactar vendedor](#)      **€ 5,72**

 <p>Motor de cepillo de dientes eléctrico para niños, potente cepillo de dientes eléctrico con vibración de 4,5x5,8mm, Micro Motor Sin Núcleo imán NdFeB 615 DC 1,5-VV        [Resumen del pedido]        € 5,09 X1        Propiedades de artículo: Spcs:   </p>	<p>Confirmación recibida</p> <p>Finalizado</p>	<p><b>Añadir a la cesta</b></p>
---	--	---------------------------------

N.º pedido: 3011114479341895 [Ver detalles](#)      Tienda: KY WIN ROBOT Score      Importe total:   
 Hora y fecha de pedido: 10:46 Apr. 21 2021      [Ver tienda](#) | [Contactar vendedor](#)      **€ 1,54**



Sombrero de paja con luz LED, 50 Uds., 5mm, diodo blanco superbrillante, 0,3 W, 0,5 W, 0,75 W, potencia FS, 0,5 W, diodo emisor de luz, rojo, amarillo, verde, azul, cálido

[Resumen del pedido]

€ 1,09 X1

Propiedades de artículo: Orange



Confirmación recibida

Finalizado

[Añadir a la cesta](#)

N.º pedido: 3011114479181895 [Ver detalles](#)      Tienda: Shenzhen Hong Ming electronics co., LTD      Importe total:   
 Hora y fecha de pedido: 10:46 Apr. 21 2021      [Ver tienda](#) | [Contactar vendedor](#)      **€ 22,44**



Conjunto de chips ATMEGA32U4 100%, (5-10 piezas), ATMEGA32U4-AU, nuevo QFP-44

[Resumen del pedido]

€ 22,44 X1

Propiedades de artículo: 3pcs



Confirmación recibida

Finalizado

[Añadir a la cesta](#)

N.º pedido: 3011114479201895 [Ver detalles](#)      Tienda: 889 Sold Store      Importe total:   
 Hora y fecha de pedido: 10:46 Apr. 21 2021      [Ver tienda](#) | [Contactar vendedor](#)      **€ 1,17**



10 unidades/lote 495SMD 16.000MHz 16MHz pasiva de oscilador de cristal SMD en Stock

[Resumen del pedido]


€ 0,64 X1



Confirmación recibida

Finalizado

[Añadir a la cesta](#)

N.º pedido: 3011114479361895 [Ver detalles](#)      Tienda: GREATZT Score      Importe total:   
 Hora y fecha de pedido: 10:46 Apr. 21 2021      [Ver tienda](#) | [Contactar vendedor](#)      **€ 9,91**



1Set SPI I2C GY-9250 MPU 9250 MPU-9250 9-eje actiud + Gyro + acelerador + Sensor magnetómetro Junta módulo MPU9250 3-5V de potencia

[Resumen del pedido]


€ 3,66 X2



Confirmación recibida

Finalizado

[Añadir a la cesta](#)

N.º pedido: 3011114479381895 [Ver detalles](#)      Tienda: Charging for your Lives' Store      Importe total:   
 Hora y fecha de pedido: 10:46 Apr. 21 2021      [Ver tienda](#) | [Contactar vendedor](#)      **€ 7,63**



20cm USB3.0 a USB3.0/2.0 USB3.0 hembra a USB Dual hombre de potencia Extra de datos Y Cable de extensión

[Resumen del pedido]

€ 2,36 X3

Propiedades de artículo: USB3.0 to USB 3.0/2.0



Confirmación recibida

Finalizado

[Añadir a la cesta](#)

N.º pedido: 3011114479401895 [Ver detalles](#)      Tienda: New IC Store      Importe total:   
 Hora y fecha de pedido: 10:46 Apr. 21 2021      [Ver tienda](#) | [Contactar vendedor](#)      **€ 1,11**



10 Uds. MIC5219-3.38M5 MIC5219 MIC5219-3.3 LM321MFX LM321 SY8008BAAC SY8008B SY8008 MIC5219-3.3YM5 RT8059GJ5 RT8059 SOT-23-5 SOT23

[Resumen del pedido]

€ 0,58 X1

Propiedades de artículo: MIC5219-3.38M5



Confirmación recibida

Finalizado

[Añadir a la cesta](#)


N.º pedido: 3011114479421895 [Ver detalles](#) | Tienda: Makeit Store | Importe total: € 0,86  
 Hora y fecha de pedido: 10:46 Apr. 21 2021 | [Ver tienda](#) | [Contactar vendedor](#)

 <p>50 Uds 1N914 diodo [Resumen del pedido] € 0,31 X1</p>	<p>Confirmación recibida</p>	<p>Finalizado</p>	<p><a href="#">Añadir a la cesta</a></p>
--	------------------------------	-------------------	--

N.º pedido: 3011114479441895 [Ver detalles](#) | Tienda: xin cheng electronic | Importe total: € 1,82  
 Hora y fecha de pedido: 10:46 Apr. 21 2021 | [Ver tienda](#) | [Contactar vendedor](#)

 <p>Interruptor deslizante 20 piezas SS12D00, Interruptor largo de manija de alta calidad, 3MM/4MM/5MM/6MM, 3 pines, 1 PZT, 2 posiciones [Resumen del pedido] € 0,64 X1 Propiedades de artículo: Handle length 6mm</p>	<p>Disputa finalizada Progreso de reembolso</p>	<p>Finalizado</p>	<p><a href="#">Añadir a la cesta</a></p>
---	---	-------------------	--











N.º pedido: 3011114479301895 [Ver detalles](#) | Tienda: Great IT electronic components co., LTD | Importe total: € 1,35  
 Hora y fecha de pedido: 10:46 Apr. 21 2021 | [Ver tienda](#) | [Contactar vendedor](#)

 <p>5 uds táctil interruptor de botón de empuje momentáneo 12*12*7,3 MM Micro interruptor de botón + 5 uds tacto Cap(5 colores) para Arduino interruptor [Resumen del pedido] € 0,43 X2 Propiedades de artículo: 5pcs black Switch</p>	<p>Confirmación recibida</p>	<p>Finalizado</p>	<p><a href="#">Añadir a la cesta</a></p>
---	------------------------------	-------------------	--



N.º pedido: 3011051903701895 [Ver detalles](#) | Tienda: FXI Electronics Co., Ltd. | Importe total: € 3,27  
 Hora y fecha de pedido: 10:36 Apr. 21 2021 | [Ver tienda](#) | [Contactar vendedor](#)

 <p>Resistor de chip SMD 100 1/4W 0R ~ 10M, 0805 Uds. 0 10R 100R 220R 330R 470R 1K 4,7 K 10K 47K 100K 0 10 100 330 470 ohm [Resumen del pedido] € 0,39 X1 Propiedades de artículo: 150R Entrega Combinada</p>	<p>Pedido cancelado</p>	<p>Finalizado</p>	<p><a href="#">Añadir a la cesta</a></p>
 <p>Resistor de chip SMD 100 1/4W 0R ~ 10M, 0805 Uds. 0 10R 100R 220R 330R 470R 1K 4,7 K 10K 47K 100K 0 10 100 330 470 ohm [Resumen del pedido] € 0,39 X1 Propiedades de artículo: 10K Entrega Combinada</p>	<p>Pedido cancelado</p>		
 <p>Resistor de chip SMD 100 1/4W 0R ~ 10M, 0805 Uds. 0 10R 100R 220R 330R 470R 1K 4,7 K 10K 47K 100K 0 10 100 330 470 ohm [Resumen del pedido] € 0,39 X1 Propiedades de artículo: 22R</p>	<p>Pedido cancelado</p>		

N.º pedido: 3011051903301895 [Ver detalles](#) Tienda: Co's zoo Store  
 Hora y fecha de pedido: 10:36 Apr. 21 2021 Ver tienda | Contactar vendedor Importe total: € 8,20

 100 Uds 0805 SMD Chip condensador cerámica multicapa 0.5PF - 47UF 22pF 100pF 1nF 100nF 0,1 uF 1uF 2,2 uF 4,7 uF 10uF 22uF 106, 104, 105 [Resumen del pedido] € 0,77 X1 Propiedades de artículo: 470PF 	Pedido cancelado	Finalizado	<a href="#">Añadir a la cesta</a>
 100 Uds 0805 SMD Chip condensador cerámica multicapa 0.5PF - 47UF 22pF 100pF 1nF 100nF 0,1 uF 1uF 2,2 uF 4,7 uF 10uF 22uF 106, 104, 105 [Resumen del pedido] € 0,77 X1 Propiedades de artículo: 22PF 	Pedido cancelado		
 100 Uds 0805 SMD Chip condensador cerámica multicapa 0.5PF - 47UF 22pF 100pF 1nF 100nF 0,1 uF 1uF 2,2 uF 4,7 uF 10uF 22uF 106, 104, 105 [Resumen del pedido] € 0,77 X1 Propiedades de artículo: 100NF 	Pedido cancelado		
 100 Uds 0805 SMD Chip condensador cerámica multicapa 0.5PF - 47UF 22pF 100pF 1nF 100nF 0,1 uF 1uF 2,2 uF 4,7 uF 10uF 22uF 106, 104, 105 [Resumen del pedido] € 1,09 X1 Propiedades de artículo: 1UF 	Pedido cancelado		
 100 Uds 0805 SMD Chip condensador cerámica multicapa 0.5PF - 47UF 22pF 100pF 1nF 100nF 0,1 uF 1uF 2,2 uF 4,7 uF 10uF 22uF 106, 104, 105 [Resumen del pedido] € 1,74 X1 Propiedades de artículo: 22UF 	Pedido cancelado		

N.º pedido: 3011114479281895 [Ver detalles](#) Tienda: CLeE LED factory Store  
 Hora y fecha de pedido: 10:46 Apr. 21 2021 Ver tienda | Contactar vendedor Importe total: € 7,10

 0,39 pulgadas rojo/azul/Verde/blanco 7 segmento 0,39 "Pantalla LED de 1/2/3/4/5 Bit tubo Digital serie pantalla led digital [Resumen del pedido] € 7,10 X1 Propiedades de artículo: 2 Bit + RED + Common anode 	Confirmación recibida	Finalizado	<a href="#">Añadir a la cesta</a>
---	-----------------------	------------	-----------------------------------

N.º pedido: 3010957604181895 [Ver detalles](#) Tienda: Estdardyn Official Store  
 Hora y fecha de pedido: 11:26 Apr. 21 2021 Ver tienda | Contactar vendedor Importe total: € 2,30

 USBASP-programador USBISP AVR, USB ISP, USB, ASP, ATMEGA8, ATMEGA128, compatible con Win7, 64K, 10 pines a 6 pines [Resumen del pedido] € 1,32 X1 Propiedades de artículo: 1ast 	Confirmación recibida	Finalizado	<a href="#">Añadir a la cesta</a>
--	-----------------------	------------	-----------------------------------

PEDIDO REALIZADO 12 de julio de 2021	TOTAL EUR 20,99	ENVIAR A Amazon Hub Locker - celestino ▾	PEDIDO N.º 171-9911240-3732326 <a href="#">Ver los detalles del pedido</a>   <a href="#">Factura ▾</a>
---	--------------------	---	---

### Entregado el 13 jul. 2021

Tu paquete se ha recogido del Amazon Locker



**PLA Filamento 1,75 mm, TINMORRY Materiales de Impresión 3D Filamento 1 kg Spool, Blanco**  
El periodo de devolución terminó el 12 ago. 2021

[Comprar de nuevo](#)

[Ver tu artículo](#)

[Localiza tu paquete](#)

[Evaluar al vendedor](#)

[Escribir una opinión sobre el producto](#)

PEDIDO REALIZADO 21 de abril de 2021	TOTAL EUR 27,57	ENVIAR A Amazon Hub Locker - rabano ▾	PEDIDO N.º 407-6037688-5817154 <a href="#">Ver los detalles del pedido</a>   <a href="#">Factura ▾</a>
---	--------------------	--	---

### Entregado el 25 abr. 2021

Tu paquete se ha recogido del Amazon Locker



**GTIWJUNG 35PCS Interruptor de Rodillo Interruptor Mini Límite Micro CNC Final de Carrera enrutador Router Impresora 3D Brazo de Palanca del Interrupto**  
El periodo de devolución terminó el 26 may. 2021

[Comprar de nuevo](#)

[Ver tu artículo](#)

[Escribir una opinión sobre el producto](#)

### Entregado el 25 abr. 2021

Tu paquete se ha recogido del Amazon Locker



**Heschen Electroimán solenoide HS-0730B DC6V 1A 5N 10 mm Carrera Tipo de empuje y tracción Marco abierto**  
El periodo de devolución terminó el 26 may. 2021

[Comprar de nuevo](#)

[Ver tu artículo](#)

[Escribir una opinión sobre el producto](#)

PEDIDO REALIZADO 17 de febrero de 2021	TOTAL EUR 56,23	ENVIAR A Amazon Hub Locker - rabano ▾	PEDIDO N.º 405-7004543-9947559 <a href="#">Ver los detalles del pedido</a>   <a href="#">Factura ▾</a>
---	--------------------	--	---



**KeeYees Pro Micro ATmega32U4 5V 16MHz Micro USB Placa de Desarrollo Módulo Microcontrolador para Arduino IDE Leonardo Bootloader (3 Piezas)**  
El periodo de devolución terminó el 21 mar. 2021

[Comprar de nuevo](#)

[Ver tu artículo](#)

[Escribir una opinión sobre el producto](#)



**AZDelivery Jumper Wire Cables de Puente 20 cm 3 x 40 pcs cada uno Macho-Hembra/Macho-Macho/Hembra-Hembra compatible con Arduino y Raspberry Pi Breadbo**  
El periodo de devolución terminó el 20 mar. 2021

[Comprar de nuevo](#)

[Ver tu artículo](#)

[Escribir una opinión sobre el producto](#)



**Arduino Micro with Headers [A000053]**  
El periodo de devolución terminó el 20 mar. 2021

[Comprar de nuevo](#)

[Ver tu artículo](#)



**AmazonBasics - Cable alargador USB 2.0 tipo A (3 m)**  
Elegibilidad de devolución ▾

[Comprar de nuevo](#)

[Ver tu artículo](#)

[Escribir una opinión sobre el producto](#)



**Juego de tornillos M3, Orthland 1110 unidades, tornillos hexagonales, juego de tornillos de acero inoxidable 304, con botón de seis cabezas, tuercas**  
El periodo de devolución terminó el 18 jul. 2021

[Comprar de nuevo](#)

[Ver tu artículo](#)

Producto	Acción del producto	Estado del pedido	Acción del pedido
ID del pedido: <b>G706996</b> <a href="#">Dirección de envío</a> Hora y fecha del pedido : 2021/5/17 23:39:33		Total del pedido: <b>\$ 25.00</b>	
Servicio: <b>Lidia Li</b>   <a href="#">Póngase en contacto con el representante de ventas (0 sin leer)</a>		Pedido completado <a href="#">Ver detalles</a>  <a href="#">Ver logística</a>	
 50x50mm 2 Capas, Grosor: 1.6 mm, Acabado de cobre: 1, Acabado superficial: HASL with lead [Nº de producto: W363490ASL3] \$ 5.00 & 5Piezas  Tiempo de preparación: 3-4 días  <a href="#">+Añadir a la Lista de deseos</a>	 pistola - CAD/CAM ZIP <a href="#">Sturesskill</a>  100% Proceso <a href="#">Repetir pedido</a> <a href="#">Disputa abierta</a>	<a href="#">Factura</a> <a href="#">Dejar comentarios</a>	

Información básica : PCB estándar

Nº de producto :	W363490ASL3	Estado del pedido :	Pedido completado
Tiempo de creación :	2021/5/17 23:26:22	Tiempo de preparación :	3-4 días
Tiempo de finalización estimado :	2021-05-21 Zona horaria de China (GMT+8)	Archivo Gerber :	pistola - CAD/CAM ZIP

Información de parámetros :

Tipo de placa :	Piezas sueltas	Forma de panel :	
Diseños diferentes en el panel :	1	Tolerar X-osit en el panel :	
Dimensiones :	50 x 50 mm	Cantidad :	5
Capa :	2 Capas	Material :	FR-4: TG130
Grosor :	1.6 mm	Min pista/espacio :	6/6mil
Min tamaño agujero :	0.3mm †	Máscara de soldadura :	Blanco
Serigrafía :	Negro	conector electrónico de borde :	No
Acabado superficial :	HASL con plomo	"HASL" por "ENIG" :	No
Procesamiento de vías :	Vías cubiertas	Acabado de cobre :	1 oz Cu
Pcb extra Número de producto :		Opciones adicionales :	Marcado UL Ninguno.
Número de PO :		Fabricación :	



## 2. Código implementado

```
#include <AbsMouse.h>//Similar a Mouse.h pero con posicionamiento
absoluto y 16 bit
#include <Keyboard.h>//Permitirá que los botones pulsados se reciban
como teclas del teclado alfanumérico
#include "MPU9250.h"//Biblioteca para calibrado y lectura de sensores
#include <SevenSeg.h>//Biblioteca para seleccionar el display
#include <SoftwareReset.h>//Reinicio por software

//Macros en las que se definen los pines físicos
#define CORREDERA 23
#define BOTON1 0
#define BOTON2 2
#define BOTON3 3
#define GATILLO 22

#define RETROCESO 10
#define VIBRACION 5

#define DISPA 7
#define DISPB 21
#define DISPC 20
#define DISPD 19
#define DISPE 18
#define DISPF 9
#define DISPG 8
#define DIGIT1 12
#define DIGIT2 4
#define DISPDP 6

//Variables propias de la parte electrónica, ACTUADORES, BOTONES Y
TEMPORIZADORES
bool corredera=true;//Los valores booleanos actuarán para poder
detectar cuando un botón se pulsa y se levanta de nuevo
bool boton1=true;
bool boton2=true;
bool boton3=true;
bool gatillo=true;
unsigned char cargador=0;
unsigned char balas=10;//Balas por defecto
unsigned long timer_engine;//se declara la variable que guardará el
tiempo de ejecución cada vez que se active el gatillo para los motores
bool modo_auto=0;//se declara la variable que guardará el tiempo de
ejecución para generar un tren de pulsos de modo similar a PWM
unsigned long timer_auto=4294967295;//Valor máximo inicial para que la
primera vez funcione sí o sí el modo automático

//Se inicializan los puertos del DISPLAY
SevenSeg disp(DISPA,DISPB,DISPC,DISPD,DISPE,DISPF,DISPG); //A B C D E
F G
const int numOfDigits=2;//Pines que seleccionan los dígitos
int digitPins[numOfDigits]={DIGIT1,DIGIT2};//Vector de dígitos

//Variables relativas al POSICIONAMIENTO
int resolucion[2]={3440,1440};//Vector que indicará la resolución del
programa (por defecto Ultra Wide)
```

```

int mouse_x_before=resolucion[0]/2;//Posición inicial del mouse, dado
que emplearemos posicionamiento absoluto
int mouse_y_before=resolucion[1]/2;

//Cálculos precisión Giroscopios Z e Y (encargados de determinar la
posición)
int pulgadas=32;//Por defecto las pulgadas son 32
int distancia=40;//por defecto la distancia son 40cm
#define PUL2CM 2.54 //una pulgada son 2.54cm
unsigned int relasp[2]={0,0};//se define la relación de aspecto
float dimensiones[2]={0,0};//se define el vector de anchura/altura de
la pantalla
float angRad[2]={0,0};//se define el vector de la apertura angular
para giroscopios, radianes (anchura/altura)
float angRad_state[2]={0,0};//Guardaremos el valor acumulado de los
ángulos en radianes (evitando salir de los márgenes)

//Inicialización biblioteca MPU9250, se instancia un objeto de la
clase, empleando el protocolo SPI y seleccionando el pin 17 como pin
de SS (Set System)
MPU9250 IMU(SPI,17);
int status;

void setup() {
    //Se establece la configuración de los pines de los botones en modo
pull-up, el de los motores en forma de salida
    pinMode(CORREDERA,INPUT_PULLUP);
    pinMode(BOTON1,INPUT_PULLUP);
    pinMode(BOTON2,INPUT_PULLUP);
    pinMode(BOTON3,INPUT_PULLUP);
    pinMode(GATILLO,INPUT_PULLUP);
    pinMode(BOTON3,INPUT_PULLUP);
    pinMode(RETROCESO,OUTPUT);
    pinMode(VIBRACION,OUTPUT);

    //Se inicializan los valores del display mediante el objeto disp
    disp.setDigitPins(numOfDigits, digitPins);
    disp.setCommonAnode(); //Indicamos que funciona como activa en baja,
podría hacerse con "setActivePinState (int segActive, int digActive)";

    //Variables de los menús
    bool inicio=true;//Variable para el inicio común
    bool men_av=false;//variable para el menú avanzado
    //Variables del menú avanzado
    unsigned char men_sel=0;
    bool men_sel_fix=false;
    unsigned char submen_sel=0;
    bool submen_sel_fix=false;

    //MENÚ INICIAL --> SELECCIÓN TAMAÑO DEL CARGADOR
    while (inicio==true) {

        disp.clearDisp();//Modo menú selección cargador o menú rápido (--)
        disp.changeDigit(0);
        disp.writeDigit('-');
        disp.changeDigit(1);
        disp.writeDigit('-');
    }
}

```

```

disp.clearDisp();

while (digitalRead(BOTON1)==LOW&&inicio==true){ //Las combinaciones
de botones permitirán seleccionar el tamaño del cargador y fijarlo
cargando el dispositivo mediante la corredera
    disp.write(balas);

    if (digitalRead(GATILLO)==LOW && gatillo ==
false){//Gatillo para añadir balas
        gatillo = true;
        if(balas<99){
            balas = balas+1;
        }
    }
    if (digitalRead(GATILLO)==HIGH && gatillo == true){
        gatillo = false;
    }

    if (digitalRead(BOTON3)==LOW && boton3 == false){//Botón 3
para eliminar balas
        boton3 = true;
        if(balas>1){
            balas = balas-1;
        }
    }
    if (digitalRead(BOTON3)==HIGH && boton3 == true){
        boton3 = false;
    }

    if (digitalRead(BOTON2)==LOW && boton2 == false){//Botón 2
para saltos grandes en munición (de 10 en 10)
        boton2 = true;
        if(balas<10){
            balas = 10;
        }
        else if (balas<20){
            balas = 20;
        }
        else if (balas<30){
            balas = 30;
        }
        else if (balas<40){
            balas = 40;
        }
        else if (balas<50){
            balas = 50;
        }
        else if (balas<60){
            balas = 60;
        }
        else if (balas<70){
            balas = 70;
        }
        else if (balas<80){
            balas = 80;
        }
        else if (balas<90){
            balas = 90;
        }
        else{

```

```

        balas=1;
    }

}

if (digitalRead(BOTON2)==HIGH && boton2 == true){
    boton2 = false;
}

if (digitalRead(CORREDERA)==LOW && corredera ==
false){//Corredera para fijar las balas del cargador
    corredera = true;
    disp.writeDigit('N');
    cargador=balas;
    inicio=false;
}
if (digitalRead(CORREDERA)==HIGH && corredera == true){
    corredera = false;
}

}

if(digitalRead(BOTON2)==LOW && digitalRead(BOTON3)==LOW){//Si se
pulsan los botones 2 y 3 simultáneamente se entra al menú avanzado
    men_av=true;
}

//MENÚ AVANZADO
while(men_av==true){
    submen_sel=0;
    submen_sel_fix=false;
    disp.clearDisp();//Modo menú avanzado (AX)
    disp.changeDigit(0);
    disp.writeDigit('A');
    disp.changeDigit(1);
    disp.writeDigit(men_sel);
    disp.clearDisp();

    if(digitalRead(BOTON3)==LOW && boton3==false){
        boton3=true;
        men_av=false;
    }
    if (digitalRead(BOTON3)==HIGH && boton3 == true){
        boton3 = false;
    }

    if(digitalRead(BOTON1)==LOW && boton1==false){
        boton1=true;
        men_sel++;
        if(men_sel>3){
            men_sel=0;
        }
    }
    if (digitalRead(BOTON1)==HIGH && boton1 == true){
        boton1 = false;
    }
    if(digitalRead(GATILLO)==LOW && gatillo == false){
        gatillo=true;
        men_sel_fix=true;
    }
    if (digitalRead(GATILLO)==HIGH && gatillo == true){
        gatillo = false;
    }
}

```

```

}
if(men_sel_fix==true){
  switch(men_sel){
  case 0://Selección de resolución
  while(submen_sel_fix==false){
    disp.clearDisp();//Modo submenú resolución
    disp.changeDigit(0);
    disp.writeDigit('R');//Resolución
    disp.changeDigit(1);
    switch(submen_sel){
      case 0:
        disp.writeDigit('H');//"HD" 1366x768(16:9)
        break;
      case 1:
        disp.writeDigit('F');//FHD 1920x1080 (16:9)
        break;
      case 2:
        disp.writeDigit('Q');//QHD 2K 2560x1440(16:9)
        break;
      case 3:
        disp.writeDigit('Y');//UltraWide 3440x1440 (21:9)
        break;
      default:
        disp.writeDigit('U');//UHD "4K" 3840x2160 (16:9)
    }
    disp.clearDisp();
    if(digitalRead(BOTON1)==LOW && boton1 == false){//Botón 1
cambia de opción
      boton1=true;
      submen_sel++;
      if(submen_sel>5){
        submen_sel=0;
      }
    }
    if (digitalRead(BOTON1)==HIGH && boton1 == true){
      boton1 = false;
    }
    if(digitalRead(GATILLO)==LOW && gatillo == false){//Gatillo
selecciona la opción
      gatillo=true;
      switch(submen_sel){
        case 0://"HD" 1366x768(16:9)
del programa
          resolucion[0]=1366;//Vector que indicará la resolución
          resolucion[1]=768;
          break;
        case 1://FHD 1920x1080 (16:9)
          resolucion[0]=1920;
          resolucion[1]=1080;
          break;
        case 2://QHD 2K 2560x1440(16:9)
          resolucion[0]=2560;
          resolucion[1]=1440;
          break;
        case 3://UltraWide 3440x1440 (21:9)
          resolucion[0]=3440;
          resolucion[1]=1440;
          break;
        default://UHD "4K" 3840x2160 (21:9)
          resolucion[0]=3840;
          resolucion[1]=2160;
      }
    }
  }
}

```

```

    }
    mouse_x_before=resolucion[0]/2;//Posición inicial del
mouse, dado que emplearemos posicionamiento absoluto
    mouse_y_before=resolucion[1]/2;
    submen_sel_fix=true;
    men_sel_fix=false;
}
if (digitalRead(GATILLO)==HIGH && gatillo == true){
    gatillo = false;
}
if(digitalRead(BOTON3)==LOW && boton3==false){//Si deseamos
salir sin cambios
    boton3=true;
    submen_sel_fix=true;
    men_sel_fix=false;
}
if (digitalRead(BOTON3)==HIGH && boton3 == true){
    boton3 = false;
}
}
break;
case 1://Selección de pulgadas
submen_sel=1;
while(submen_sel_fix==false){

    if(submen_sel<10){
        disp.clearDisp();//Modo submenú pulgadas
        disp.changeDigit(0);
        disp.writeDigit('P');//Pulgadas
        disp.changeDigit(1);
        disp.writeDigit(submen_sel);
    }
    else{
        disp.write(submen_sel);
    }
    disp.clearDisp();

    if(digitalRead(BOTON1)==LOW && boton1 == false){//Cambiar
entre opciones
        boton1=true;
        submen_sel++;
        if(submen_sel>99){ //Máximo de 99 pulgadas
            submen_sel=1;
        }
    }
    if (digitalRead(BOTON1)==HIGH && boton1 == true){
        boton1 = false;
    }
    if(digitalRead(BOTON2)==LOW){//Incrementa el valor de forma
rápida
        submen_sel++;
        if(submen_sel>99){
            submen_sel=1;
        }
    }
    if(digitalRead(GATILLO)==LOW && gatillo == false){
//Selecciona la opción
        gatillo=true;
        pulgadas=submen_sel;
        submen_sel_fix=true;
        men_sel_fix=false;
}

```

```

    }
    if (digitalRead(GATILLO)==HIGH && gatillo == true){
        gatillo = false;
    }

    if(digitalRead(BOTON3)==LOW && boton3==false){//Si deseamos
salir sin cambios
        boton3=true;
        submen_sel_fix=true;
        men_sel_fix=false;
    }
    if (digitalRead(BOTON3)==HIGH && boton3 == true){
        boton3 = false;
    }
}
break;
case 2://Distancia a la que se situará el jugador
submen_sel=1;
while(submen_sel_fix==false){
    disp.clearDisp();//Modo submenú distancia en cm
    disp.changeDigit(0);
    disp.writeDigit('D');//Distancia
    disp.changeDigit(1);
    disp.writeDigit(submen_sel);// indicará 1 para 10cm, 2 para
20cm... (Por defecto 40cm)

    disp.clearDisp();
    if(digitalRead(BOTON1)==LOW && boton1 == false){//El botón 1
cambia entre las opciones
        boton1 = true;
        submen_sel++;
        if(submen_sel>9){
            submen_sel=1;
        }
    }
    if (digitalRead(BOTON1)==HIGH && boton1 == true){
        boton1 = false;
    }
    if(digitalRead(GATILLO)==LOW && gatillo == false){//El
gatillo selecciona la opción deseada
        gatillo=true;
        distancia=submen_sel*10;
        submen_sel_fix=true;
        men_sel_fix=false;
    }
    if (digitalRead(GATILLO)==HIGH && gatillo == true){
        gatillo = false;
    }
}
if(digitalRead(BOTON3)==LOW && boton3==false){//Si deseamos
salir sin cambios
        boton3=true;
        submen_sel_fix=true;
        men_sel_fix=false;
    }
    if (digitalRead(BOTON3)==HIGH && boton3 == true){
        boton3 = false;
    }
}
break;

case 3://Disparo automático

```

```

submen_sel=0;
while(submen_sel_fix==false){
    disp.clearDisp();//Modo submenú disparo automático
    disp.changeDigit(0);
    disp.writeDigit('-');//Ráfaga
    disp.changeDigit(1);
    disp.writeDigit(submen_sel);

    disp.clearDisp();
    if(digitalRead(BOTON1)==LOW && boton1 == false){//Alternar
entre las opciones sí o no
        boton1 = true;
        submen_sel++;
        if(submen_sel>1){
            submen_sel=0;
        }
    }
    if (digitalRead(BOTON1)==HIGH && boton1 == true){
        boton1 = false;
    }
    if(digitalRead(GATILLO)==LOW && gatillo ==
false){//Selecciona la opción
        gatillo=true;
        modo_auto=submen_sel;
        submen_sel_fix=true;
        men_sel_fix=false;
    }
    if (digitalRead(GATILLO)==HIGH && gatillo == true){
        gatillo = false;
    }
    if(digitalRead(BOTON3)==LOW && boton3==false){//Si deseamos
salir sin cambios
        boton3=true;
        submen_sel_fix=true;
        men_sel_fix=false;
    }
    if (digitalRead(BOTON3)==HIGH && boton3 == true){
        boton3 = false;
    }
}

} //FIN switch() submenú
} //FIN Inicio submenú

} //Fin menú avanzado
} //Fin menú selección cargador o menú rápido

```

```

AbsMouse.init(resolucion[0], resolucion[1]); //Se inicializa la
resolución llamando a la biblioteca AbsMouse e iniciando así su
funcionamiento

```

```

//POSICIONAMIENTO
//Se realizan los cálculos de optimización del posicionamiento, en
función de los valores seleccionados por el usuario
//Calculo de ángulo en radianes respecto posición en pantalla
//inicialmente se calcula la relación de aspecto solo habrá 2-->
21:9 y 16:9 (viene ya recogido en la variable relasp)
if((resolucion[0]/resolucion[1])>=2){

```



```

    relasp[0]=21;
    relasp[1]=9;
}
else{
    relasp[0]=16;
    relasp[1]=9;
}

//Se aplica Pitágoras para conocer el tamaño de la pantalla
float lado;
lado=sqrt(pow((pulgadas*PUL2CM),2)/((pow(relasp[0],2)+(pow(relasp[1],
2)))));
dimensiones[0]=relasp[0]*lado;//anchura
dimensiones[1]=relasp[1]*lado;//altura

//Una vez tenemos las dimensiones de la pantalla será necesario
calcular el ángulo máximo de barrido respecto al usuario
//para ello volveremos a aplicar Pitágoras en función de la distancia
de separación de usuario

//Angulo de barrido horizontal
float h;
h=sqrt(pow((dimensiones[0]/2),2)+(pow(distancia,2)));
angRad[0]=2*(asin((dimensiones[0]/2)/h));//Asin ya da el resultado en
radianes

//Angulo de barrido vertical
h=sqrt(pow((dimensiones[1]/2),2)+(pow(distancia,2)));
angRad[1]=2*(asin((dimensiones[1]/2)/h));

//Ahora ya poseemos los datos del ángulo máximo que podrá barrer el
usuario.
//Se ha observado que el sensor se lee aproximadamente cada 13ms
*****de media, con lo cual será necesario dividir los rad/s
entre los milisegundos que tarda en leerse para obtener los radianes
//Ahora realizaremos una cuantificación de los píxeles de la pantalla
a los valores intermedios de los ángulos en radianes
//para cada iteración del bucle map(value, fromLow, fromHigh, toLow,
toHigh) se empleará algo similar a esta función pero implementado
manualmente
angRad[0]=angRad[0]*(1000/11);//tarda 11ms ES NECESARIO AJUSTARLO EN
FUNCIÓN DEL TIEMPO DE EJECUCIÓN DEL CÓDIGO DEL BUCLE, PARA PASARLO A
RAD (o *1/0.013)
angRad[1]=angRad[1]*(1000/11);
angRad_state[0]=(angRad[0]/2);
angRad_state[1]=(angRad[1]/2);

//FINALMENTE ESPERAMOS A UNA PRESIÓN DEL GATILLO, EN LA CUAL SE COLOCA
EL CURSOR EN EL CENTRO DE LA PANTALLA Y POSTERIORMENTE SE INICIA LA
COMUNICACIÓN CON LA BIBLIOTECA MPU9250 CALIBRANDO ASÍ LOS SENSORES
while(digitalRead(GATILLO)==HIGH){
    AbsMouse.move(resolucion[0]/2,resolucion[1]/2);
    disp.clearDisp();//Modo calibración, será importante mantener la
postura apuntando al centro de la pantalla mientras se calibra el
mando(CA)
    disp.changeDigit(0);
    disp.writeDigit('C');
}

```

```

    disp.changeDigit(1);
    disp.writeDigit('A');
    disp.clearDisp();
} //Si no presionas el gatillo se queda en bucle

//Cuando se inicie el calibrado la pistola vibrará. Cuando la
vibración finalice, el dispositivo estará listo para utilizarse.
digitalWrite(VIBRACION, HIGH);

    // start communication with IMU
    status = IMU.begin();//Calibrará automáticamente el sensor con la
posición actual
    if (status < 0) {
        while(1) {}//En el caso de que se den errores el programa se
detendrá
    }
    //selección de los rangos de escala completa
    //Giroscopio
IMU.setGyroRange(MPU9250::GYRO_RANGE_2000DPS);//seleccionamos el rango
de valores del giroscopio en +/-2000grados/seg con el fin de obtener
más precisión
    //Acelerómetro
IMU.setAccelRange(MPU9250::ACCEL_RANGE_8G);// +/-2G
    //Magnetómetro --> únicamente admite +/- 4800uT (No se empleará)

    //Selección del ancho de banda del DLPF (Digital Low Pass Filter)
IMU.setDlpfBandwidth(MPU9250::DLPF_BANDWIDTH_5HZ);//Bajamos el ancho
de banda del filtro desde 184 (Valor por defecto), hasta 5, con el fin
de suavizar las variaciones.

Keyboard.begin();//Se inicializa la detección de los botones como
teclas de un teclado HID convencional

//Se para la vibración cuando se termine de inicializar IMU
digitalWrite(VIBRACION, LOW);

}

void loop() {

    IMU.readSensor();//Inicialmente se establece la lectura del sensor
para cada ciclo

    disp.write (balas);//El display estará activo actualizándose

    if (digitalRead(CORREDERA)==LOW && corredera ==
false){//Actualización del cargador en función de la corredera, cada
vez que se activa, la corredera rellena nuevamente el cargador de
balas
        corredera = true;
        balas=cargador;
        Keyboard.write('r');//Se asigna la tecla r a la acción de recargar
(comúnmente predefinida "reload")
    }
    if (digitalRead(CORREDERA)==HIGH && corredera == true){
        corredera = false;
    }
}

```

```

//Control de posición y establecimiento de gestos
  if (digitalRead(BOTON1)==HIGH){//Si el botón de gestos está activado
ya no moveremos el ratón (BOTON1) y dejaremos de tener en cuenta los
valores del sensor
    if(abs(IMU.getGyroZ_rads())>0.05 && angRad_state[1]<=angRad[1] &&
angRad_state[1]>=0 ){//Evitamos que se mueva ante pequeñas
variaciones, y si nos salimos de los márgenes no realizará movimientos
en pantalla
    AbsMouse.move(mouse_x_before,(angRad_state[1]-
IMU.getGyroZ_rads())*(resolucion[1]/angRad[1]));//modo suave de mayor
precisión
    mouse_y_before=(angRad_state[1]-
IMU.getGyroZ_rads())*(resolucion[1]/angRad[1]);
    }
    if(abs(IMU.getGyroZ_rads())>0.05){//Si nos salimos de los márgenes
de la pantalla se seguirán contando los movimientos, aunque el puntero
no se mueva
    angRad_state[1]=angRad_state[1]-IMU.getGyroZ_rads();
    }

    if(abs(IMU.getGyroY_rads())>0.05 && angRad_state[1]<=angRad[0] &&
angRad_state[0]>=0 ){//Se procede de la misma manera para el eje
vertical
    AbsMouse.move((((angRad_state[0]-
IMU.getGyroY_rads())*(resolucion[0]/angRad[0]))),mouse_y_before);
    mouse_x_before= (((angRad_state[0]-
IMU.getGyroY_rads())*(resolucion[0]/angRad[0])));
    }
    if(abs(IMU.getGyroY_rads())>0.05){
    angRad_state[0]=angRad_state[0]-IMU.getGyroY_rads();
    }

}else{//En caso de que se presione el botón de gestos, el cursor
volverá a situarse en el centro de la pantalla y en función de los
movimientos se simularán botones
    AbsMouse.move(resolucion[0]/2,resolucion[1]/2); //Cursor en el
centro de la pantalla
    mouse_x_before=resolucion[0]/2;
    mouse_y_before=resolucion[1]/2;
    angRad_state[0]=(angRad[0]/2);
    angRad_state[1]=(angRad[1]/2);

    //Se simulará la cruceta del teclado (asdw) mediante gestos
    //Movimiento lateral
    if (IMU.getAccelZ_mss())<-7){
    Keyboard.write('d');
    }
    if (IMU.getAccelZ_mss())>7){
    Keyboard.write('a');
    }

    //Movimiento horizontal
    if (IMU.getAccelX_mss())<-4){
    Keyboard.write('s');
    }
    if (IMU.getAccelX_mss())>4){
    Keyboard.write('w');
    }
}

```

```

}

//Accionamiento del gatillo
if (digitalRead(GATILLO)==LOW && gatillo == false && balas>0 &&
modo_auto==0){//Si se aprieta el gatillo en modo semiautomático
gatillo = true;
balas=balas-1;//Al accionar el gatillo se restan las balas
AbsMouse.press();//Solo se activa el clic del gatillo si quedan
balas en el cargador
timer_engine = millis(); //Se toma nota del tiempo de ejecución
cuando se aprieta el gatillo (se pone la condición fuera del if del
gatillo apretado)
digitalWrite(VIBRACION, HIGH);
digitalWrite(RETROCESO, HIGH);
}

if(millis()-timer_engine>100 && modo_auto==0){//tras 100ms se
desactiva la vibración en modo semiautomático
digitalWrite(VIBRACION, LOW);
digitalWrite(RETROCESO, LOW);
}

if (digitalRead(GATILLO)==HIGH && gatillo == true){
gatillo = false;
AbsMouse.release();//Se suelta el clic
}

//Accionamiento del gatillo en modo automático (simulará un pulso PWM)
if (digitalRead(GATILLO)==LOW && modo_auto==1 && balas>0){//Si se
aprieta el gatillo, está en modo automático y hay balas
if(millis()-timer_auto>200) {//Se accionarán los motores tras cada
ciclo de 200ms
timer_engine = millis();//Periodo de vibración de 200ms
timer_auto=millis();//Tiempo de vibración
digitalWrite(VIBRACION, HIGH);
digitalWrite(RETROCESO, HIGH);
AbsMouse.press();//Se efectúa un clic y se suelta
AbsMouse.release();
balas=balas-1;
}
}
if(millis()-timer_engine>50 && modo_auto==1){//tras 50ms se
desactiva la vibración para el modo automático
digitalWrite(VIBRACION, LOW);
digitalWrite(RETROCESO, LOW);
}

//Accionamiento de los botones
if (digitalRead(BOTON2)==LOW && boton2 == false){//Botón de acción 1
(Botón 2)
boton2 = true;
Keyboard.write((char) 32);//Se asigna la barra espaciadora a la
acción de saltar (comúnmente predefinida "space")
}
if (digitalRead(BOTON2)==HIGH && boton2 == true){
boton2 = false;
}
}

```

```

if (digitalRead(BOTON3)==LOW && boton3 == false){//Botón de acción 2
(Botón 3)
  boton3 = true;
  Keyboard.write('z');//Se asigna la tecla z
}
if (digitalRead(BOTON3)==HIGH && boton3 == true){
  boton3 = false;
}

//RESETEAR LA PISTOLA POR SI NECESITAS VOLVER A ESTABLECER PARÁMETROS
DEL MENÚ SIN TENER QUE ENCHUFAR DE NUEVO EL DISPOSITIVO
if( digitalRead(BOTON1)== LOW && digitalRead(BOTON2)== LOW &&
digitalRead(BOTON3)== LOW && digitalRead(GATILLO)== LOW &&
digitalRead(CORREDERA)== LOW && digitalRead(BOTON1)== LOW ){
  //Software reset
  softwareReset::standard() ; // Reset using the standard method reset
using watchdog timer ( full-reset - recommended ) [ha sido necesario
cambiar a dot_a_linkage=false en el fichero library.properties]
}

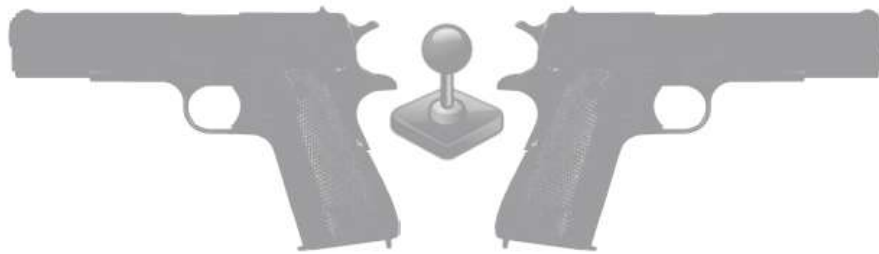
}

```

### 3. Manual de usuario

# MANUAL DE USUARIO

## GUN CONTROLLER V1



### PARTES



## **CORREDERA**

**EL BOTÓN DE LA CORREDERA PODRÁ ACCIONARSE AL DESPLAZAR HACIA EL USUARIO LA PARTE MÓVIL SUPERIOR DEL DISPOSITIVO.**

**EN EL MODO DE JUEGO PERMITIRÁ RECARGAR LAS BALAS DEL CARGADOR, TANTO SI ESTE ESTÁ LLENO, COMO SI ESTÁ VACÍO, PARA ELLO ACCIONARÁ LA TECLA “R” EN EL EQUIPO DEL MISMO MODO QUE LO HARÍA UN TECLADO.**

**\*ES IMPORTANTE QUE LA CORREDERA RECUPERE SU POSICIÓN INICIAL, EN CASO DE “ENCASQUILLAMIENTO” AYUDE A LA MISMA A RECUPERAR SU LUGAR MANUALMENTE.**

## **GATILLO**

**EL BOTÓN DEL GATILLO PODRÁ ACCIONARSE PULSANDO EL GATILLO DEL DISPOSITIVO HACIA LA EMPUÑADURA.**

**EN EL MODO DE JUEGO PERMITIRÁ EFECTUAR LA ACCIÓN DE DISPARO, EFECTUANDO UN CLIC IZQUIERDO EN EL CURSOR DE LA PANTALLA.**

**\*EN CASO DE MANTENER PULSADO EL GATILLO EN EL MODO DE JUEGO, ÚNICAMENTE SE REALIZARÁ UN DISPARO, ADEMÁS NO SE EFECTUARÁ LA ACCIÓN DE SOLTAR EL CLIC IZQUIERDO HASTA QUE EL USUARIO NO CESE SU PULSACIÓN. (SIEMPRE QUE NO SE ENCUENTRE EN EL MODO AUTOMÁTICO)**

---

## **BOTÓN 1**

**EL BOTÓN 1 PODRÁ ACCIONARSE PULSANDO EL BOTÓN SUPERIOR.**

**EN EL MODO DE JUEGO PERMITIRÁ PASAR AL MODO DE GESTOS, ASÍ COMO ANULAR TEMPORALMENTE EL MOVIMIENTO DEL CURSOR EN LA PANTALLA MIENTRAS ESTÉ ACCIONADO.**

**\*EN CASO DE DEJAR DE PULSAR ESTE BOTÓN, SERÁ NECESARIO QUE EL USUARIO APUNTE A LA POSICIÓN CENTRAL DE LA PANTALLA, DADO QUE EL CURSOR SE SITUARÁ EN EL CENTRO DE LA MISMA CADA VEZ QUE SE ACCIONE EL BOTÓN DE GESTOS. DE OTRA FORMA, EL APUNTADOR SUFRIRÁ ERRORES A LA HORA DE DETERMINAR LA POSICIÓN EN PANTALLA.**

## **BOTÓN 2**

**EL BOTÓN 2 PODRÁ ACCIONARSE PULSANDO EL BOTÓN CENTRAL.**

**EN EL MODO DE JUEGO PERMITIRÁ ACCIONAR LA BARRA ESPACIADORA EN EL EQUIPO DEL MISMO MODO QUE LO HARÍA UN TECLADO.**

**LA TECLA SE PODRÁ CONFIGURAR EN FUNCIÓN DEL JUEGO EMPLEADO (TÍPICAMENTE EMPLEADA PARA SALTAR).**

## **BOTÓN 3**

**EL BOTÓN 3 PODRÁ ACCIONARSE PULSANDO EL BOTÓN INFERIOR.**

**EN EL MODO DE JUEGO PERMITIRÁ ACCIONAR LA TECLA “Z” EN EL EQUIPO, DEL MISMO MODO QUE LO HARÍA UN TECLADO.**

**LA TECLA SE PODRÁ CONFIGURAR EN FUNCIÓN DEL JUEGO EMPLEADO.**

## **INDICADOR LED**

**EL INDICADOR LED APORTARÁ INFORMACIÓN AL USUARIO.**

**SERVIRÁ PARA ESTABLECER LA CONFIGURACIÓN INICIAL TRAS CONECTAR EL DISPOSITIVO.**

**DURANTE EL MODO DE JUEGO MOSTRARÁ LAS BALAS RESTANTES DEL CARGADOR EN TIEMPO REAL.**

---

## **PUERTO USB**

**EL PUERTO USB PERMITIRÁ ALIMENTAR EL DISPOSITIVO, ASÍ COMO ESTABLECER LA COMUNICACIÓN ENTRE EL MISMO Y EL EQUIPO AL QUE SE CONECTE.**

**PARA CONECTAR EL DISPOSITIVO AL EQUIPO SERÁ NECESARIO EMPLEAR EL CABLE USB INCORPORADO EN EL PAQUETE.**

## **CONECTOR DOBLE USB**

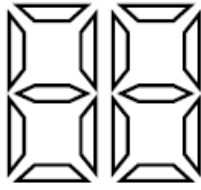
**EL CONECTOR DOBLE USB PERMITIRÁ EMPLEAR EL DISPOSITIVO A PLENO RENDIMIENTO.**

**SE RECOMIENDA EMPLEARLO EN DOS PUERTOS USB 3.0 O SUPERIOR, EN SU DEFECTO UN PUERTO USB 3.0 O SUPERIOR Y OTRO USB 2.0. TAMBIÉN PERMITE SU CONEXIÓN CON DOS PUERTOS USB 2.0.**

**LA UTILIZACIÓN DEL CONECTOR DOBLE USB NO ES INDISPENSABLE PARA PODER EMPLEAR EL DISPOSITIVO, PERO PODRÍA MEJORAR LA EXPERIENCIA DE USUARIO.**



# SÍMBOLOS



INDICADOR LED APAGADO



INDICADOR LED ENCENDIDO



CORREDERA, GATILLO Y BOTONES NO PULSADOS



CORREDERA, GATILLO Y BOTONES PULSADOS

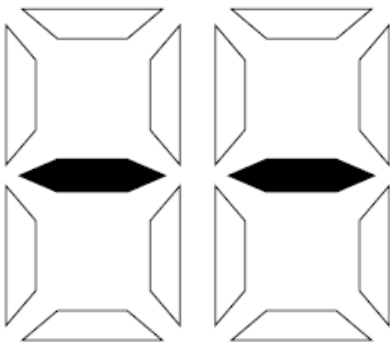
## INSTRUCCIONES DE USO

TRAS ADQUIRIR EL DISPOSITIVO GUN CONTROLLER V1 CONECTE EL CONECTOR DOBLE USB A DOS PUERTOS USB 3.0, EN CASO DE NO SER POSIBLE CONSULTE LA DESCRIPCIÓN DEL CONECTOR DOBLE USB.

POSTERIORMENTE SERÁ NECESARIO CONECTAR EL CONECTOR DOBLE USB AL CABLE USB INCORPORADO Y ESTE ÚLTIMO AL DISPOSITIVO GUN CONTROLLER V1. (EN CASO DE NO EMPLEAR EL CONECTOR DOBLE USB CONECTAR UN EXTREMO DEL CABLE DIRECTAMENTE AL EQUIPO)

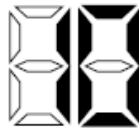
UNA VEZ CONECTADO EL INDICADOR LED MOSTRARÁ EL MENÚ INICIAL (-).

## MENÚ INICIAL



PERMITE VISUALIZAR EL TAMAÑO DEL CARGADOR ESTABLECIDO SI SE MANTIENE PULSADO, ADEMÁS PERMITE MODIFICAR ESTE VALOR SI SE ACCIONA JUNTO CON EL GATILLO Y LOS BOTONES 2 Y 3.

VALOR POR DEFECTO CARGADOR: 10



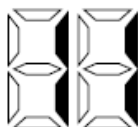
SI MIENTRAS SE MANTIENE EL BOTÓN 1 SE ACCIONA EL BOTÓN 2, EL TAMAÑO DEL CARGADOR SE INCREMENTARÁ DE 10 EN 10.

VALOR POR DEFECTO TRAS UNA PULSACIÓN: 20



SI MIENTRAS SE MANTIENE EL BOTÓN 1 SE ACCIONA EL GATILLO, EL TAMAÑO DEL CARGADOR SE INCREMENTARÁ DE 1 EN 1.

VALOR POR DEFECTO TRAS UNA PULSACIÓN: 11



SI MIENTRAS SE MANTIENE EL BOTÓN 1 SE ACCIONA EL BOTÓN 3, EL TAMAÑO DEL CARGADOR SE DECREMENTARÁ DE 1 EN 1.

VALOR POR DEFECTO TRAS UNA PULSACIÓN: 9

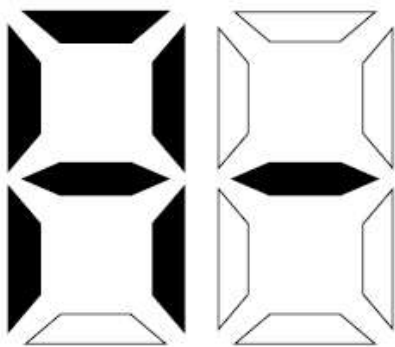


SI MIENTRAS SE MANTIENE EL BOTÓN 1 SE ACCIONA LA CORREDERA, EL TAMAÑO DEL CARGADOR QUEDARÁ FIJADO Y SE PASARÁ AUTOMÁTICAMENTE A LA FASE DE CALIBRACIÓN (CA).

VALOR POR DEFECTO TRAS UNA PULSACIÓN: CA

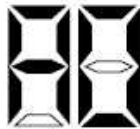


# MENÚ AVANZADO



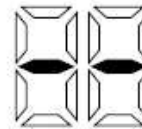
PARA PODER ACCEDER AL MENÚ AVANZADO SERÁ NECESARIO PULSAR DE FORMA SIMULTÁNEA LOS BOTONES 2 Y 3 DESDE EL MENÚ INICIAL, NO SERÁ NECESARIO MANTENERLOS.

VALOR POR DEFECTO MENÚ AVANZADO: A0



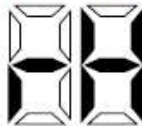
UNA VEZ SE ACCEDA AL MENÚ AVANZADO SE PODRÁ EMPLEAR EL BOTÓN 3 EN CUALQUIER MOMENTO PARA RETROCEDER AL PASO ANTERIOR SIN GUARDAR LOS CAMBIOS. PERMITIRÁ VOLVER AL MENÚ INICIAL.

VALOR POR DEFECTO TRAS UNA PULSACIÓN: VUELTA AL MENÚ



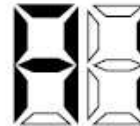
UNA VEZ SE ACCEDA AL MENÚ AVANZADO SE PODRÁ EMPLEAR EL GATILLO EN CUALQUIER MOMENTO PARA SELECCIONAR LOS SUBMENÚS Y APLICAR LOS CAMBIOS DE CONFIGURACIÓN.

VALOR POR DEFECTO TRAS UNA PULSACIÓN: SELECCIÓN A0



UNA VEZ SE ACCEDA AL MENÚ AVANZADO SE PODRÁ EMPLEAR EL BOTÓN 1 EN CUALQUIER MOMENTO PARA ROTAR ENTRE LAS OPCIONES Y SUBMENÚS.

VALOR POR DEFECTO TRAS UNA PULSACIÓN: SELECCIÓN A1



## SUBMENÚS



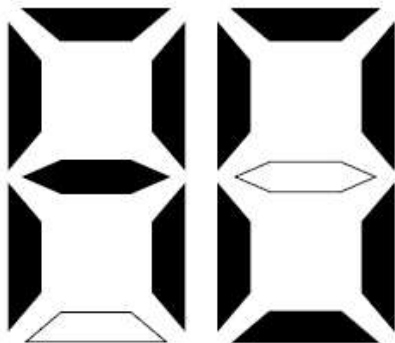
SELECCIÓN DE RESOLUCIÓN A0

SELECCIÓN DE PULGADAS A1

SELECCIÓN DE DISTANCIA A2

SELECCIÓN MODO AUTOMÁTICO A3

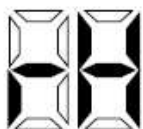
# RESOLUCIÓN



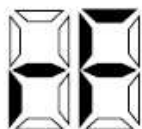
EN EL MENÚ AVANZADO, EL SUBMENÚ RESOLUCIÓN PERMITE MODIFICAR LA RESOLUCIÓN (POR DEFECTO 3440X1440) DEL DISPOSITIVO CON LA FINALIDAD DE PODER SER EMPLEADO EN DISTINTAS PANTALLAS.

A CONTINUACIÓN, SE MOSTRARÁN LAS DISTINTAS OPCIONES DE CONFIGURACIÓN PARA ESTE SUBMENÚ.

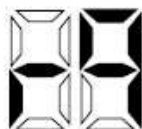
\* EN CASO DE REINICIAR EL DISPOSITIVO LAS MODIFICACIONES LLEVADAS A CABO EN EL MENÚ AVANZADO DEJARÁN DE SURTIR EFECTO Y SERÁ NECESARIO VOLVER A APLICARLAS.



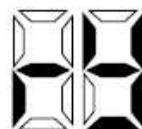
RH: RESOLUCIÓN "HD" (HIGH DEFINITION) 1366X768(16:9)



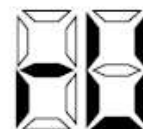
RF: RESOLUCIÓN FHD (FULL HD) 1920X1080 (16:9)



RQ: RESOLUCIÓN QHD (QUAD HD) 2K 2560X1440(16:9)

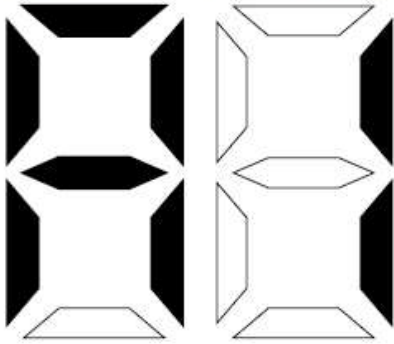


RY: RESOLUCIÓN UW (ULTRA WIDE) 3440X1440 (21:9)



RU: RESOLUCIÓN UHD (ULTRA HD) "4K" 3840X2160 (16:9)

# PULGADAS



EN EL MENÚ AVANZADO, EL SUBMENÚ PULGADAS PERMITE MODIFICAR EL TAMAÑO DE LA PANTALLA EMPLEADA, (POR DEFECTO 32”), DEL DISPOSITIVO CON LA FINALIDAD DE PODER SER EMPLEADO EN DISTINTAS PANTALLAS.

A CONTINUACIÓN SE MOSTRARÁN LAS DISTINTAS OPCIONES DE CONFIGURACIÓN PARA ESTE SUBMENÚ.

\*EN CASO DE REINICIAR EL DISPOSITIVO LAS MODIFICACIONES LLEVADAS A CABO EN EL MENÚ AVANZADO DEJARÁN DE SURTIR EFECTO Y SERÁ NECESARIO VOLVER A APLICARLAS.

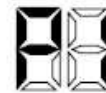


DENTRO DEL MENÚ AVANZADO EN EL SUBMENÚ PULGADAS (A1), EL BOTÓN 1 PERMITIRÁ ROTAR DE UNO EN UNO EL VALOR DE LAS PULGADAS, AL IGUAL QUE LAS OPCIONES DEL MENÚ AVANZADO.



DENTRO DEL MENÚ AVANZADO EN EL SUBMENÚ PULGADAS (A1), EL BOTÓN 2 PERMITIRÁ ROTAR DE MANERA RÁPIDA EL VALOR DE LAS PULGADAS EN FUNCIÓN DEL TIEMPO QUE SE MANTENGA PULSADO.

## RANGO

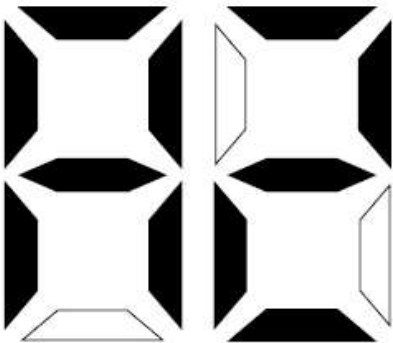


VALOR INICIAL  
(MÍNIMO)



VALOR MÁXIMO  
(99)

# DISTANCIA



EN EL MENÚ AVANZADO, EL SUBMENÚ DISTANCIA PERMITE MODIFICAR LA DISTANCIA A LA PANTALLA (POR DEFECTO 40CM) DEL DISPOSITIVO, CON LA FINALIDAD DE PODER ADAPTARSE A DISTINTAS SITUACIONES.

A CONTINUACIÓN SE MOSTRARÁN LAS DISTINTAS OPCIONES DE CONFIGURACIÓN PARA ESTE SUBMENÚ.

\*EN CASO DE REINICIAR EL DISPOSITIVO LAS MODIFICACIONES LLEVADAS A CABO EN EL MENÚ AVANZADO DEJARÁN DE SURTIR EFECTO Y SERÁ NECESARIO VOLVER A APLICARLAS.

A LA HORA DE CONFIGURAR LA DISTANCIA DESDE EL DISPOSITIVO AL CENTRO DE LA PANTALLA SERÁ NECESARIO ESTABLECER LA DISTANCIA EN CM.

PARA ELLO EL SUBMENÚ DISTANCIA MUESTRA LAS DECENAS EN CENTÍMETROS DE LAS POSIBLES DISTANCIAS QUE SE PUEDEN CONFIGURAR.

\*LAS DISTANCIAS ESTÁN CALCULADAS RESPECTO AL CENTRO DE LA PANTALLA, EN LÍNEA RECTA, SUPONIENDO QUE TANTO LA PANTALLA COMO EL DISPOSITIVO SE ENCUENTRAN PERPENDICULARES AL SUELO.

## RANGO

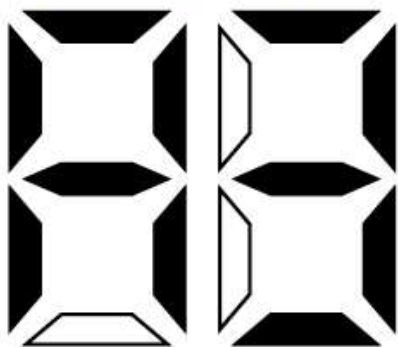


VALOR INICIAL  
(MÍNIMO 10CM)



VALOR MÁXIMO  
(90CM)

# MODO AUTOMÁTICO DE DISPARO



EN EL MENÚ AVANZADO, EL SUBMENÚ MODO AUTOMÁTICO DE DISPARO PERMITE MODIFICAR EL TIPO DE DISPARO (POR DEFECTO SEMIAUTOMÁTICO) DEL DISPOSITIVO CON LA FINALIDAD DE PODER EMULAR METRALLETAS Y OTROS SUBFUSILES DE JUEGOS.

A CONTINUACIÓN SE MOSTRARÁN LAS DISTINTAS OPCIONES DE CONFIGURACIÓN PARA ESTE SUBMENÚ.

\* EN CASO DE REINICIAR EL DISPOSITIVO LAS MODIFICACIONES LLEVADAS A CABO EN EL MENÚ AVANZADO DEJARÁN DE SURTIR EFECTO Y SERÁ NECESARIO VOLVER A APLICARLAS.

EL TIPO DE DISPARO INFLUIRÁ DE MANERA DIRECTA EN LA EXPERIENCIA DE USUARIO.

AL IGUAL QUE EN LAS ARMAS REALES, UN MODO DE DISPARO AUTOMÁTICO EFECTUARÁ MÚLTIPLES DISPAROS ÚNICAMENTE MANTENIENDO PULSADO EL GATILLO.

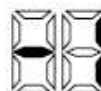
POR OTRO LADO, UN MODO MANUAL/SEMIAUTOMÁTICO ÚNICAMENTE EFECTUARÁ UN DISPARO AUNQUE EL GATILLO SE MANTENGA PULSADO.

PARA ESTABLECER EL MODO MANUAL, BASTARÁ CON SELECCIONAR UNA TAMAÑO DE CARGADOR IGUAL A 1 BALA.

## VALORES

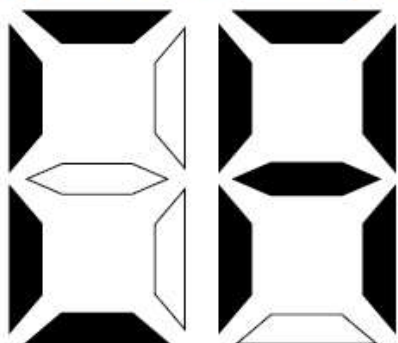


DISPARO MANUAL/  
SEMIAUTOMÁTICO



DISPARO AUTOMÁTICO

# CALIBRACIÓN



TRAS HABER APLICADO LAS CONFIGURACIONES DESEADAS TANTO EN EL MENÚ AVANZADO COMO EN EL MENÚ INICIAL, SE ACTIVARÁ LA CORREDERA PARA CONFIRMAR LAS BALAS FIJADAS EN EL CARGADOR MIENTRAS SE PULSA EL BOTÓN 1.

A CONTINUACIÓN EL INDICADOR LED MOSTRARÁ QUE EL DISPOSITIVO SE ENCUENTRA EN LA FASE DE CALIBRACIÓN (CA).

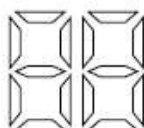
\* UNA MALA CALIBRACIÓN DEL DISPOSITIVO PODRÍA PROVOCAR FALLOS EN LA POSICIÓN DEL PUNTERO.



UNA VEZ EN LA FASE DE CALIBRACIÓN SERÁ NECESARIO APUNTAR CON EL DISPOSITIVO AL CENTRO DE LA PANTALLA Y PULSAR EL GATILLO PARA CONTINUAR.



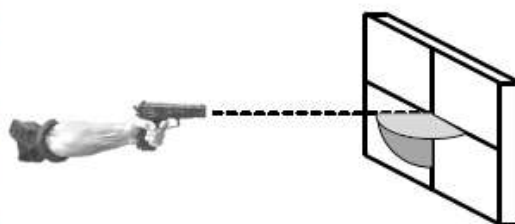
TRAS ACCIONAR EL GATILLO EL INDICADOR SE APAGARÁ.



LA CALIBRACIÓN DEBE SER LO MÁS EXACTA POSIBLE. EL USUARIO DEBERÁ COLOCAR EL DISPOSITIVO PERPENDICULAR AL SUELO Y APUNTANDO EN LÍNEA RECTA AL CENTRO DE LA PANTALLA. SE RECOMIENDA AYUDARSE DE LA MIRA INCORPORADA EN LA CÁRCASA.

UNA VEZ ACCIONADO EL GATILLO SE MANTENDRÁ LA POSICIÓN MIENTRAS DURE LA VIBRACIÓN DEL DISPOSITIVO. CUANDO EL DISPOSITIVO ESTÉ LISTO EL INDICADOR LED VOLVERÁ A ENCENDERSE AUTOMÁTICAMENTE Y SE DETENDRÁ LA VIBRACIÓN

## POSICIÓN CORRECTA



ÁNGULOS DE 90°

# MODO DE JUEGO

UNA VEZ LA CALIBRACIÓN SE HAYA COMPLETADO Y LA VIBRACIÓN CESE, NOS ENCONTRAREMOS EN EL MODO DE JUEGO.

EN ESTE MODO, EL CURSOR DEL EQUIPO AL QUE SE CONECTE SEGUIRÁ NUESTROS MOVIMIENTOS ANGULARES MIENTRAS NO SE ACCIONE EL BOTÓN DE GESTOS.

EN CASO DE PERCIBIR UNA FALTA DE PRECISIÓN APRETAR EL BOTÓN DE GESTOS Y VOLVER A SITUAR EL CURSOR EN LA PANTALLA, SERVIRÁ PARA RECALIBRAR DE FORMA RÁPIDA EL ARMA.

## INDICADOR LED

EN EL CASO DEL INDICADOR LED, MOSTRARÁ CONTINUAMENTE LAS BALAS QUE NOS QUEDAN EN EL CARGADOR.

ESTE NÚMERO SE IRÁ DECREMENTANDO SEGÚN SE EFECTÚEN DISPAROS.

POR OTRA PARTE, AL RECARGAR EL ARMA, EL CARGADOR RESETEARÁ SU NÚMERO TOTAL DE BALAS.

## COMPORTAMIENTO DE LOS BOTONES



TECLA "R"

CLIC IZQUIERDO MOUSE

1

TECLAS "ASDW" MEDIANTE GESTOS

2

TECLA "SPACE"

3

TECLA "Z"

DURANTE EL MODO DE JUEGO TODOS LOS BOTONES DEL DISPOSITIVO SERÁN RECONOCIDOS COMO ENTRADAS DE TECLADO O PROVENIENTES DEL RATÓN.

ESTO PERMITIRÁ ASIGNAR CONTROLES ESPECÍFICOS EN EL MENÚ DE DISTINTOS JUEGOS, MODIFICANDO LOS CARACTERES DEL TECLADO QUE DETERMINAN LAS ACCIONES.

EL BOTÓN DE GESTOS NO EFECTUARÁ POR SI SOLO NINGUNA ACCIÓN NI ENTRADA Y TENDRÁ QUE COMBINARSE CON MOVIMIENTOS PARA APORTAR LAS TECLAS DESEADAS.

## GESTOS



UNA VEZ SE ACCEDA AL MODO DE JUEGO SE PODRÁ EMPLEAR EL BOTÓN 1 EN CUALQUIER MOMENTO PARA HABILITAR LA ENTRADA DE GESTOS.

EL BOTÓN 1 NO PRODUCIRÁ NINGUNA ENTRADA AL ACCIONARSE SI EL DISPOSITIVO SE ENCUENTRA EN POSICIÓN DE CALIBRACIÓN.

PARA EFECTUAR ENTRADAS DE TECLADO DEBERÁN REALIZARSE LOS MOVIMIENTOS DESCRITOS A CONTINUACIÓN.

UNA VEZ EN EL MODO DE JUEGO PODREMOS HABILITAR LA FUNCIÓN DE GESTOS. PARA ELLO TENDREMOS QUE LLEVAR A CABO UNA SERIE DE MOVIMIENTOS ESPECÍFICOS QUE PROVOCARÁN QUE SE PRODUZCAN ENTRADAS DE TECLADO CONTINUAS.

EL ACCIONAMIENTO DEL BOTÓN DE GESTOS PRODUCIRÁ QUE EL CURSOR SE COLOQUE EN EL CENTRO DE LA PANTALLA DE FORMA INSTANTÁNEA.

\*ES IMPORTANTE QUE NO SE SUELTE EL BOTÓN DE GESTOS MIENTRAS SE EFECTÚAN LOS MOVIMIENTOS.

\*SERÁ NECESARIO APUNTAR AL CENTRO DE LA PANTALLA CUANDO SE SUELTE EL BOTÓN DE GESTOS, O SE PODRÍAN PRODUCIR FALLOS EN LA POSICIÓN DEL CURSOR

## ENTRADAS Y MOVIMIENTOS



TECLA "A"



TECLA "D"



TECLA "S"



TECLA "W"

# REINICIO

SI NOS ENCONTRAMOS EN EL MODO DE JUEGO Y DESEAMOS MODIFICAR ALGUNO DE LOS PARÁMETROS DE LA CONFIGURACIÓN INICIAL, PODREMOS REINICIAR EL DISPOSITIVO SIN NECESIDAD DE DESENCHUFARLO Y ENCHUFARLO.

PARA ELLO BASTARÁ CON PRESIONAR SIMULTÁNEAMENTE TODOS LOS BOTONES QUE INCORPORA EL ARMA.

COMO CONSECUENCIA DEL REINICIO SE ELIMINARÁN TODAS LAS CONFIGURACIONES ANTERIORMENTE APLICADAS.



---

# DISFRUTA DE TU PARTIDA