

# Controlador PWM: Diseño de Placa Base y Gabinete

Introducción al Diseño Asistido por Computadora (86.70) - Final Integrador - FIUBA  
Martin Klöckner ([mklockner@fi.uba.ar](mailto:mklockner@fi.uba.ar)) - 9 Diciembre 2024

## Introducción

En el presente trabajo se diseña la placa base y el gabinete de un controlador modulado por ancho de pulsos. Para el diseño de ambos componentes se utilizan programas de computadora, en particular KiCAD para el diseño y simulación de la placa base y FreeCAD para el diseño del gabinete y modelos 3D de componentes de la placa. Al final del trabajo se indica como obtener los archivos y un costo aproximado de fabricación.

## Esquemático y Simulación

El circuito se trata de un controlador modulado por ancho de pulsos (controlador PWM en adelante) el diseño del circuito se obtuvo del post<sup>[1]</sup> del usuario Creative Creator en el sitio web [hackster.io](https://hackster.io).

En la figura 1 a continuación se muestra el esquemático completo dibujado en KiCAD siguiendo el diseño mencionado previamente, además se agrega una fuente de alimentación de 12 volts, B1, junto con un LED y una resistencia como carga, D3 y R3 respectivamente, esto para simular el circuito, como se muestra en la directiva de simulación de spice, la cual indica una simulación de tipo transitoria con intervalo de tiempo de 10 us y un total de muestra de 1 ms.

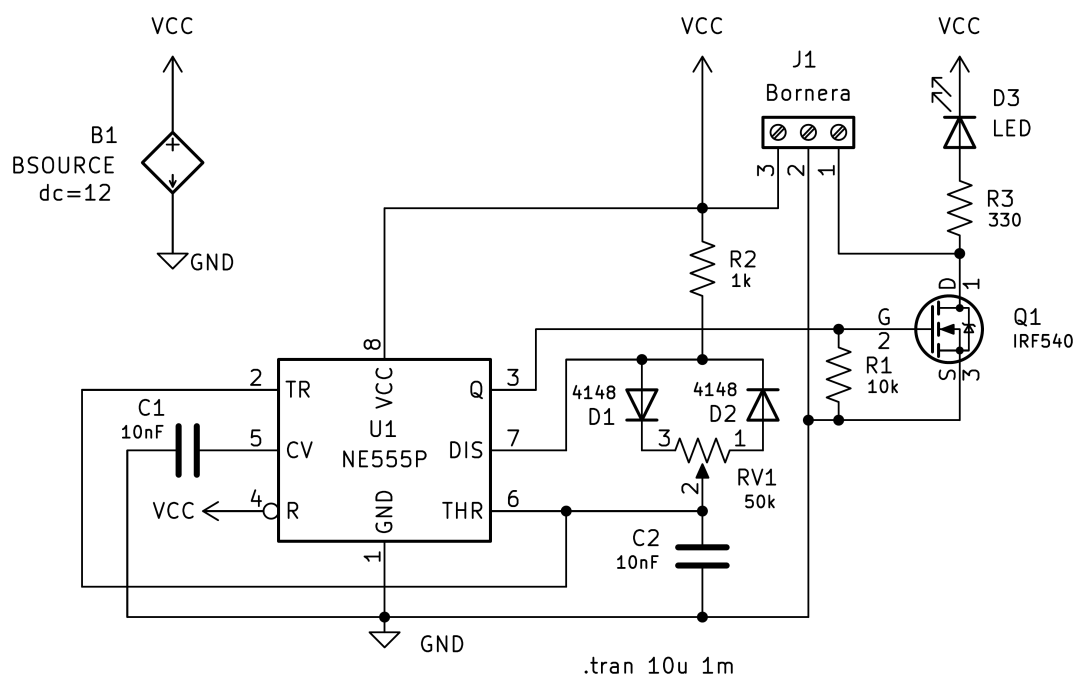


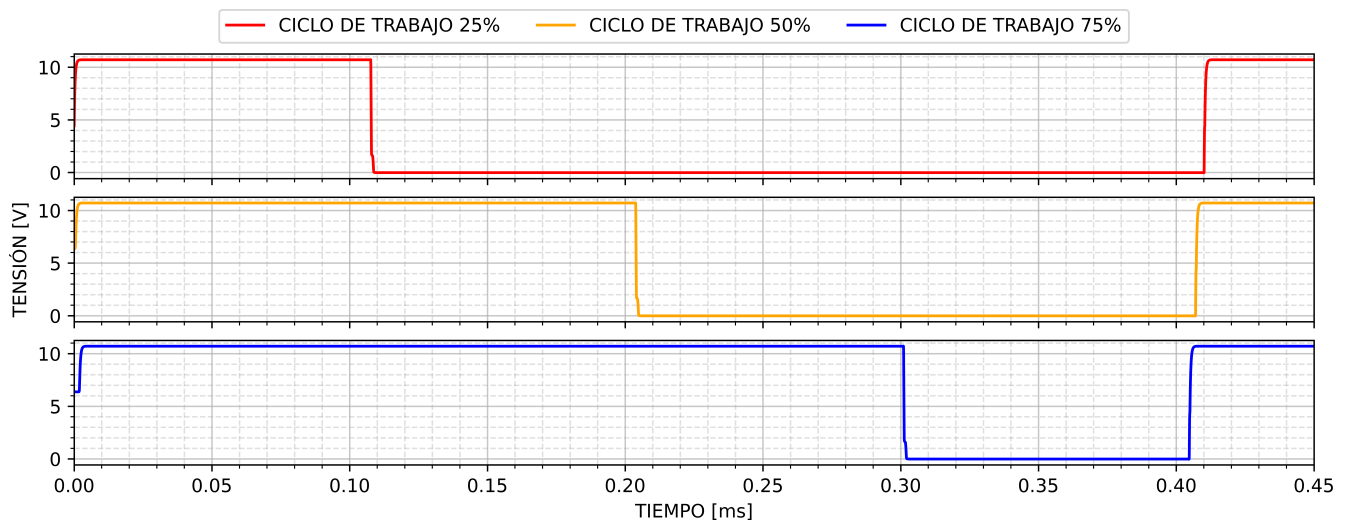
Figura 1: Circuito diseñado en KiCAD

El circuito se trata de alimentar un MOSFET Canal-N con la señal proveniente de un timer 555, el cual genera una señal PWM cuyo ciclo de trabajo es controlado por el potenciómetro RV1.

La salida del timer 555 se conecta al terminal Gate del MOSFET, el terminal Source se conecta a tierra, y el terminal Drain se conecta al negativo de la carga, de esta manera cuando el MOSFET esté en corte la carga no recibe alimentación y por ende permanecerá apagada, por el contrario cuando el MOSFET esté encendido, circulará corriente y la carga se encenderá.

<sup>[1]</sup> Creative Creator. (2020, Mar 17). LED Dimmer Circuit with 555 Timer. [hackster.io](https://www.hackster.io/dhritimanhb2015/led-dimmer-circuit-with-555-timer-bc30f7). <https://www.hackster.io/dhritimanhb2015/led-dimmer-circuit-with-555-timer-bc30f7>

En la figura 2, se muestra la salida del timer 555 cuando se varía la posición del potenciómetro RV1, se puede ver como cambia el ciclo de trabajo de la señal, es decir, se controla el tiempo en que la señal permanece activa en un período, el máximo siendo la señal activa durante todo el período, y el mínimo, la señal inactiva o apagada (aunque en la practica no se llega a una señal totalmente apagada/encendida).



**Figura 2: Señal generada por el timer 555 para diferentes posiciones del potenciómetro RV1**

El timer 555 genera una señal PWM de frecuencia relativamente constante, como se puede ver en la figura 2 en la cual aún variando el ciclo de trabajo el periodo sigue siendo el mismo: aproximadamente 0.405 ms, lo que equivale a una frecuencia de 2469 Hz. Esta frecuencia fija es controlada por el capacitor C2, la resistencia R2 y el valor total del potenciómetro RV1; siendo el calculo para la frecuencia el siguiente:

$$f \approx (5 \cdot (R1 + RV1) \cdot C2)^{-1}$$

Con los valores elegidos, 1 kΩ para R1, 50kΩ para RV1 y 10nF para C2, la frecuencia teórica aproximada resulta:

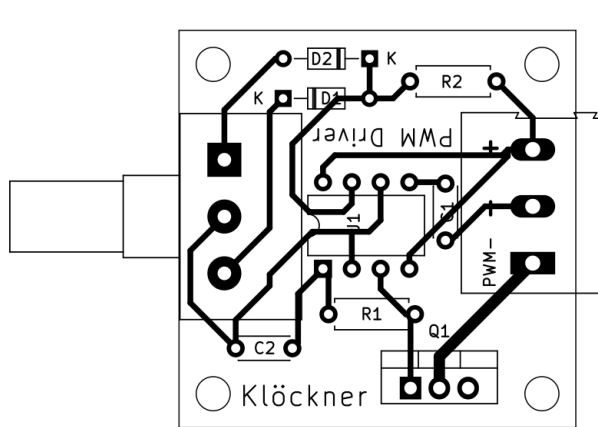
$$f \approx (5 \cdot 51k\Omega \cdot 10nF)^{-1} \Rightarrow f \approx 2222Hz$$

Teniendo en cuenta que la frecuencia es aproximadamente 2222 Hz, la carga se enciende y apaga muy rápido, utilizando un diodo LED como carga, y variando el ciclo de trabajo de la señal se puede ver como el LED parece variar su brillo en función del ciclo de trabajo, esto es porque el LED se enciende y apaga muy rápido, más rápido de lo que los ojos humanos pueden percibir, y al variar el ciclo de trabajo varía la cantidad del tiempo que el LED permanece encendido en un período. Este controlador PWM también puede ser utilizado para variar la velocidad de motores de corriente continua de igual forma en que se varía la intensidad del LED, encendiendo y apagando la fuente muchas veces por segundo y controlando el porcentaje de tiempo que permanece encendido en un período de la señal.

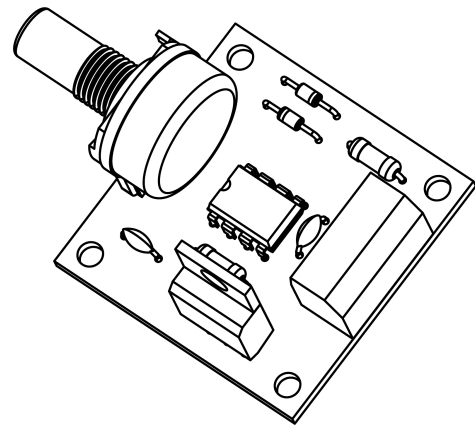
## Diseño de Placa Base

En la figura 3 se muestra una vista de capas (figura 3.1) y en 3D (figura 3.2) de la placa base del circuito diseñada utilizando la herramienta de edición de PCBs de KiCAD, cabe destacar que en la vista de capas se ha ocultado el plano a tierra, y la placa en sí, en donde se colocan los componentes, tiene opacidad nula, de manera que se aprecien mejor las pistas.

Para la mayoría de las pistas se ha utilizado un ancho de 20 mils, solo en la pista de salida (pin central del MOSFET Q1) se ha utilizado un diámetro de aproximadamente 40 mils (1mm) puesto que por ahí circula la corriente de la carga, la cual puede llegar a valores muy altos, en principio limitado por los parámetros máximos de la fuente y el MOSFET Q1.



3.1. Vista de capas

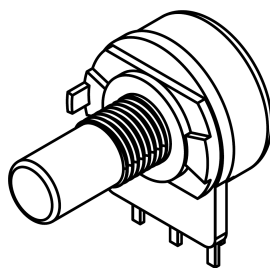


3.2. Vista 3D

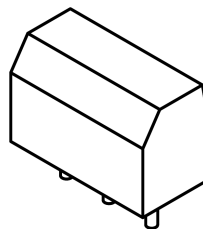
**Figura 3: Placa base diseñada utilizando KiCAD**

Para la vista en 3D de la placa, se han utilizado los modelos disponibles en KiCAD, a excepción de los modelos del potenciómetro, el transistor de paquete TO-220 y la bornera, debido a que, en el caso del potenciómetro y la bornera no se incluyen en la librería estándar de KiCAD, en cuanto al paquete TO-220 se utilizó una versión modificada del modelo incluido en la librería estándar de KiCAD, en esta versión modificada se cortan los pines de modo que se asemeje a una versión del mismo ya montado y soldado en la placa.

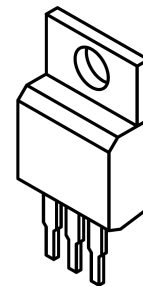
En la figura 4 se pueden ver vistas isométricas de los modelos propios diseñados en FreeCAD. En la figura 4.1 se muestra el modelo del potenciómetro, en la figura 4.2 se muestra el modelo de la bornera y en la figura 4.3, se muestra la versión modificada del paquete TO-220.



4.1. Potenciómetro



4.2. Bornera



4.3. TO-220

**Figura 4: Vistas de los modelos 3D propios**

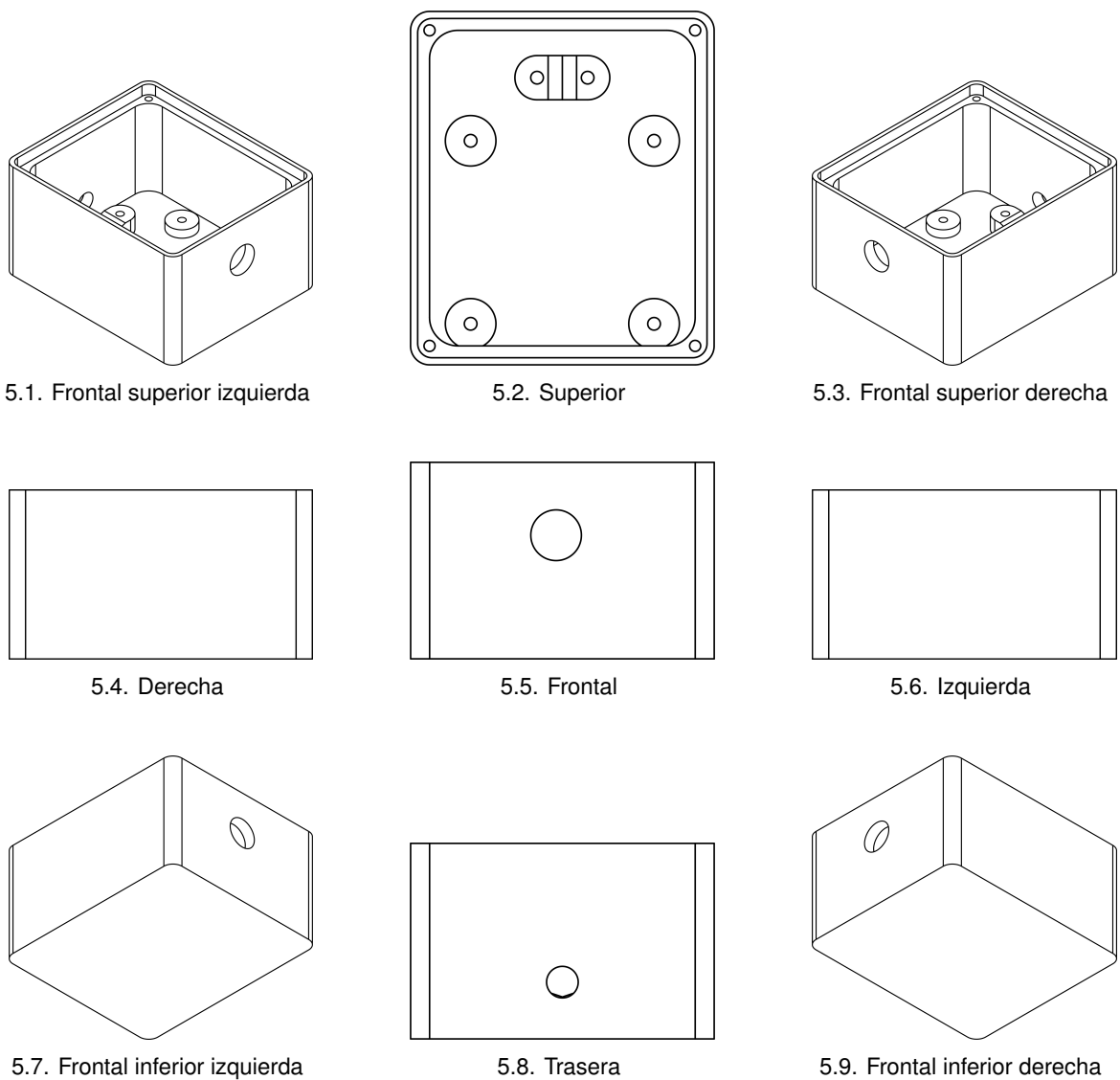
En ambos diseños propios (potenciómetro y bornera) se puede ver que el nivel de detalle no es el máximo, ya que por ejemplo, en la bornera faltarían los tornillos, o en el potenciómetro faltaría modelar la soldadura de los pines con la placa del mismo; se optó por este nivel de detalle ya que como veremos es suficiente para modelar el gabinete que contendrá la placa base. En caso de querer un mayor detalle, es más conveniente recurrir a librerías de terceros en caso de que estén disponibles (por ejemplo el diseño de bornera<sup>[2]</sup> de Daniel Školník).

## Diseño de Gabinete

Para el diseño del gabinete se exporta la vista en 3D de la placa base de KiCAD, y se importa en FreeCAD, teniendo el modelo 3D de la placa base en FreeCAD, se comienza a diseñar el gabinete en base a las dimensiones de la placa, intentando minimizar las dimensiones del mismo e intentando que el proceso de ensamble sea sencillo y además, teniendo en cuenta el tiempo de impresión del mismo.

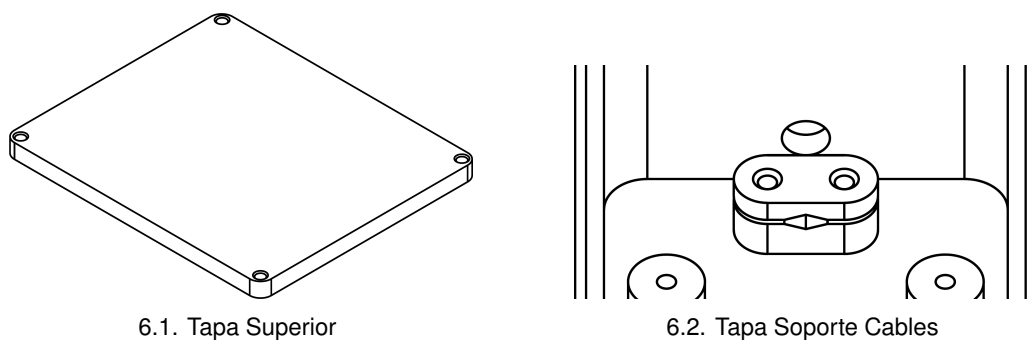
En la figura 5 se muestra una serie completa de vistas del gabinete ya finalizado, excluyendo la tapa superior del gabinete, y la tapa superior del soporte para cables, los cuales se muestran en la figura 6.

<sup>[2]</sup> Daniel Školník (2021, Nov 16), Screw terminal DG-301 5mm, *grabcad.com*. <https://grabcad.com/library/screw-terminal-dg-301-5mm-1>



**Figura 5: Vistas sin tapa del gabinete modelado en FreeCAD**

Las dimensiones del gabinete visto de frente (figura 5.5) son 48 mm de ancho, 31 mm de alto y 56 mm de profundidad; para las paredes y la tapa superior del gabinete se utilizó 3 mm de espesor, con estas dimensiones el tiempo de impresión resulta de 68 minutos según SuperSlicer (software similar a ultimaker CURA), utilizando la impresora 3D Prusa i3 MK2S con extrusor de 0.4 mm, altura de capa de 0.35 mm y densidad de relleno de 100%.



**Figura 6: Vistas de tapas del gabinete**

En la figura 6 se muestran las tapas que no se muestran en la figura 5, las cuales son la tapa superior que cierra al gabinete (figura 6.1) y la tapa que completa el soporte de cables (figura 6.2).

## Construcción de la Placa Base

Para la fabricación de la placa base se utilizó el servicio de mecanizado CNC de una placa de cobre FR4 que ofrece el Laboratorio de Circuitos Impresos (LCI) de FIUBA, los componentes se compraron en una tienda de electrónica y luego se soldaron a mano. Para la fabricación del gabinete se utilizó el servicio de impresión 3D que ofrece Beeld 3D<sup>[3]</sup>, la impresora y los parámetros de impresión utilizados se desconocen.

### Materiales y Costo de Fabricación

En la tabla 1 a continuación se muestra la lista de materiales necesarios para el armado del controlador y se incluye una lista de precios de referencia de cada componente obtenidos de la tienda Microelectrónica Componentes.<sup>[4]</sup> Al final se muestra el precio total, el cual resulta \$13.674,0 pesos argentinos, o el equivalente en dolares siendo aproximadamente \$13,0.

Componente	Cantidad	Precio [ARS \$]
Capacitor cerámico 10 nF 50 V	2	90,4
Diodo Rápido 1N4148	2	34,4
Bornera triple	1	1.416,0
MOSFET Canal-N IRF 1407	1	1.823,3
Resistencia 330 kΩ 0.5 W	1	87,3
Resistencia 10 kΩ 0.5 W	1	87,3
Resistencia 1 kΩ 0.5 W	1	87,3
Potenciómetro 100 kΩ 16 mm	1	1.132,9
Temporizador NE555P	1	687,5
Impresión de Gabinete	1	4.200,0
Mecanizado de PCB	1	4.030,0
<b>Total</b>		<b>13.676,4</b>

**Tabla 1: Precios de referencia y total de los componentes necesarios para armar el controlador**

En caso de querer producir en cantidad el controlador (por ejemplo para la venta) se puede reducir mucho el costo de producción por unidad ya que es común que las tiendas hagan descuentos por pedidos mayoristas, tanto en los componentes electrónicos como en la impresión y mecanizado del gabinete y la placa base respectivamente.

## Consideraciones para la próxima versión

Luego del ensamble final del controlador se encontraron algunos inconvenientes: en primer lugar al girar el potenciómetro RV1 en sentido horario se disminuye el ciclo de trabajo de la señal producida por el timer 555, esto es contra-intuitivo, ya que por lo general en sentido horario se incrementa la intensidad (piense en el control de volumen de un radio), una solución rápida a esto es invertir el sentido de los diodos D1 y D2. Otro inconveniente detectado utilizando un LED como carga, es la inestabilidad que se produce en la salida cuando el ciclo de trabajo de la señal es bajo, esto puede deberse a que cuando la salida del timer 555 está apagada, el terminal Drain del MOSFET queda flotando, una solución a esto es agregar una resistencia de pull-down entre los terminales 1 y 2 de la bornera J1. Por ultimo, al momento del ensamble se detectó también que el resistor R2 colisiona en parte con el soporte de uno de los tornillos de la placa base (figura 3.1, orificio superior derecho), la solución para la próxima versión sería alejar la resistencia R2 del orificio y/o hacer las paredes del soporte más finas.

<sup>[3]</sup> Beeld 3D <https://beeld.com.ar/>

<sup>[4]</sup> Microelectrónica Componentes SRL <https://www.microelectronicash.com/>

## Organización de Archivos

Los archivos del controlador se pueden obtener del repositorio de github.<sup>[5]</sup> Luego de obtener los archivos, una vista de árbol del directorio se puede ver a continuación:

```
./
├── LICENSE
├── README.md
├── informe.pdf
├── pwm_driver_case/
│   ├── pwm_driver_3d_model.FCStd
│   ├── pwm_driver_case_main_body.stl
│   ├── pwm_driver_case_main_body_cable_support_top.stl
│   └── pwm_driver_case_main_body_lid.stl
├── pwm_driver_exploded_assembly.mkv
└── pwm_driver_kicad/
    ├── pwm_driver_3d_models/
    ├── pwm_driver_kicad.kicad_pcb
    ├── pwm_driver_kicad.kicad_pro
    ├── pwm_driver_kicad.kicad_sch
    └── pwm_driver_spice_simulation_models/
```

5 directorios, 11 archivos

Los principales directorios siendo `pwm_driver_case` y `pwm_driver_kicad`. El primero refiere al gabinete, en el cual se encuentra el proyecto de FreeCAD `pwm_driver_3d_model.FCStd`, el segundo directorio: `pwm_driver_kicad`, refiere al proyecto de KiCAD, incluyendo las simulaciones y el diseño de la placa base, en el subdirectorio `pwm_driver_3d_models` se encuentran los modelos propios para las vistas 3D de la placa base y en `pwm_driver_spice_simulation_models/` se encuentran los modelos de spice para realizar las simulaciones.

En la mayoría de proyectos de FreeCAD, siendo el mayor el gabinete, se utilizó un paradigma paramétrico utilizando el banco de trabajo “spreadsheet”, el cual permite crear hojas de calculo cuyas celdas pueden ser referenciadas luego, obteniendo así un diseño facil de manipular, es decir, en caso de querer cambiar las longitudes del gabinete, por ejemplo, simplemente basta con cambiar el valor de la celda a la cual hace referencia la longitud del gabinete.

## Conclusión

En este trabajo hemos visto el proceso de diseño y fabricación de un dispositivo electrónico, desde el circuito “en papel” hasta el producto finalizado listo para ser utilizado y/o instalado.

Utilizamos en su totalidad herramientas de código abierto (open-source) que no requieren de una licencia para ser utilizadas, si bien en algunos casos se encuentran errores o fallos, hay que tener presente que son de código abierto, por lo que es fácil reportar errores a los desarrolladores. De todas maneras, se comprueba que estas herramientas de codigo abierto pueden ser utilizados y *se utilizan* en producción, en particular KiCAD.

En cuanto al controlador, el coste de producción es muy elevado para la funcionalidad que ofrece, hay que tener en cuenta que el diseño del circuito y el tipo de componentes utilizados es “antiguo” ya que a día de hoy se ha disminuido el uso de componentes through-hole y se utiliza en mayor medida componentes de montaje superficial (SMD) debido a que permiten reducir mucho el costo de fabricación. En cuanto a la funcionalidad, este tipo de controladores, por ejemplo en el caso de controlar la intensidad de un LED, tiene tendencia hacia el uso de microcontroladores y dispositivos conectados a una red de computadoras, debido a que ofrecen la posibilidad de controlar el circuito a distancia, o ampliar las funcionalidades del circuito a través de programación, utilizando la misma placa base y/o gabinete.

<sup>[5]</sup> Martin Klöckner, Pulse-width Modulation Driver, Repositorio de GitHub, [https://github.com/mjklOCKNER/pwm\\_driver](https://github.com/mjklOCKNER/pwm_driver)