

TFG

DISEÑO E IMPLEMENTACIÓN DE UN INVERSOR VARIADOR DC-AC

Titulación: Grado en Ingeniería Electrónica,
Industrial y Automática

Autor: Héctor Herrera Laorden

Tutor: Jonay Tomás Toledo Carrillo

Curso: 2016/2017

Convocatoria: Junio 2017

ÍNDICE

1- INTRODUCCIÓN GENERAL	1
1.1- Objetivo del trabajo final de grado.	1
1.2- Antecedentes.	1
1.3 Abstract.	2
1.4- Estructura.	3
1.5- Herramientas utilizadas para para el diseño del proyecto.	4
1.5.1- Herramientas utilizadas en el montaje y medición.	4
1.5.1.1-Protoboard.	4
1.5.1.2-Osciloscopio.	4
1.5.1.3-Multímetro.	5
1.5.1.4- Fuente de alimentación.....	5
1.5.1.5 Generador de funciones.	5
1.5.2- Herramientas de software.	6
1.5.2.1- LTSpice IV.....	6
1.5.2.2- Arduino IDE v1.8.2.....	6
1.5.2.3- Libre office.....	6
2- CARACTERÍSTICAS DE DISEÑO	7
2.1- Tipos de inversores.....	7
2.2- Conversor DC-DC.....	8
2.3- Puente en H.....	9
2.4- Driver del Puente en H.	12
2.5- Variador de voltaje y frecuencia.	13
3- PROGRAMACIÓN DEL ARDUINO	15
3.1- Características de Arduino.	15
3.1.1- Características de Arduino Uno.....	15
3.1.2- Funcionamiento Arduino IDE.	15
3.2- Descripción del código.	16
3.3- Comprobación de funcionamiento.	23
4- IMPLEMENTACIÓN	24
4.1- Implementación de Elevador DC-DC.....	24
4.2- Implementación de Driver del Puente en H.....	26
4.2.1- Frecuencia de la portadora en la Spwm.....	26

4.2.2- Simulación de Arduino Uno.....	28
4.2.3- Elevador de la señal Spwm.....	29
4.3- Implementación de Puente en H.....	34
4.4- Experimentos realizados.	45
5- PRESUPUESTO	50
5.1- Presupuesto de los circuitos.....	50
5.1.1 Circuito DC-DC	50
5.1.2- Circuito Driver.	50
5.1.3- Circuito puente en H y componentes adicionales.....	51
5.2- Mano de obra	51
5.3- Coste total.	51
6- CONCLUSIÓN	52
6.1- Comparativa comercial.....	52
6.2- Opinión personal	52
7- REFERENCIAS Y BIBLIOGRAFÍA.	54
ANEXOS.....	55
PLANOS, ESQUEMAS Y GRÁFICAS.....	55
HOJAS DE DATOS	88

1- INTRODUCCIÓN GENERAL.

1.1- Objetivo del trabajo final de grado.

El objetivo de este trabajo de fin de grado es el de diseñar un inversor DC-AC para el compresor del sistema de frenado neumático del vehículo autoguiado Verdino, desarrollado en la Universidad de La Laguna.

Dado que buscamos una comparativa con los inversores comerciales se tendrá en cuenta los siguientes objetivos secundarios:

- Tener un rendimiento aceptable en el inversor.
- La forma de la señal de salida se ha de adecuar al estándar comercial.
- Diseño en lazo cerrado.
- Precio competitivo.

Como un objetivo adicional se implementará un variador que permita al sistema tener un arranque suave.

1.2- Antecedentes.

El vehículo Verdino se alimenta con 6 baterías de 6 voltios cada una, por lo que para este proyecto se pretende tener una fuente de alimentación de 36 voltios.

El sistema de frenado del vehículo es un sistema por aire comprimido a 4 bares que encenderá el compresor cuando detecte que la presión en el tanque de aire disminuye por debajo de la presión establecida, y apagará nuestro sistema cuando se vuelva a alcanzar dicha medida.

El compresor precisa de una señal alterna de 220 voltios eficaces para su correcto funcionamiento y no necesita una reacción inmediata, por lo que se podrá conectar el compresor cuando el inversor haya llegado a un régimen constante y reducir su consumo en el arranque.

El desarrollo de este proyecto se llevará a cabo en el laboratorio, se simulará el funcionamiento con fuentes de alimentación y generadores de funciones.

El proyecto se diseñó para dos alumnos. Este fue dividido en dos, donde cada uno haría una parte del mismo. La parte correspondiente al otro alumno; Néstor Gómez Tejera, ya fue realizada y evaluada en el pasado curso 2015/2016.



Figura 1: Fotografia vehículo Verdino (Recuperada de Google+)

1.3 Abstract.

This project's goal is to design and implement a power inverter that is capable to transform 36V continuous current into 220V RMS (Root Mean Square) alternating current. This inverter is meant to enable a compressor the use of direct current batteries as power supply.

This project includes a design of a DC-DC elevator from 36V to 310V as well as an H Bridge to transform DC to AC. This also involves the designing of drivers for each circuit, which includes the programming of an Arduino UNO to produce SPWM (Sinusoidal Pulse with Modulation).

Furthermore, a variator is included in order to reduce power spikes at the system's start.

This project was started by other student and its continued in this document using a fragment of his work to improve it. The result is a cheaper device that can be used as intended up to 220 V RMS. It only has been tested successfully up to 110 V RMS.

Key Words: Inverter, Elevator, DC-DC, H Bridge, Arduino, Spwm, Variator.

1.4- Estructura.

La estructura del inversor seguirá el esquema siguiente donde las baterías de 36 Voltios alimentan los sistemas secundarios tal como se muestra en la figura 2.

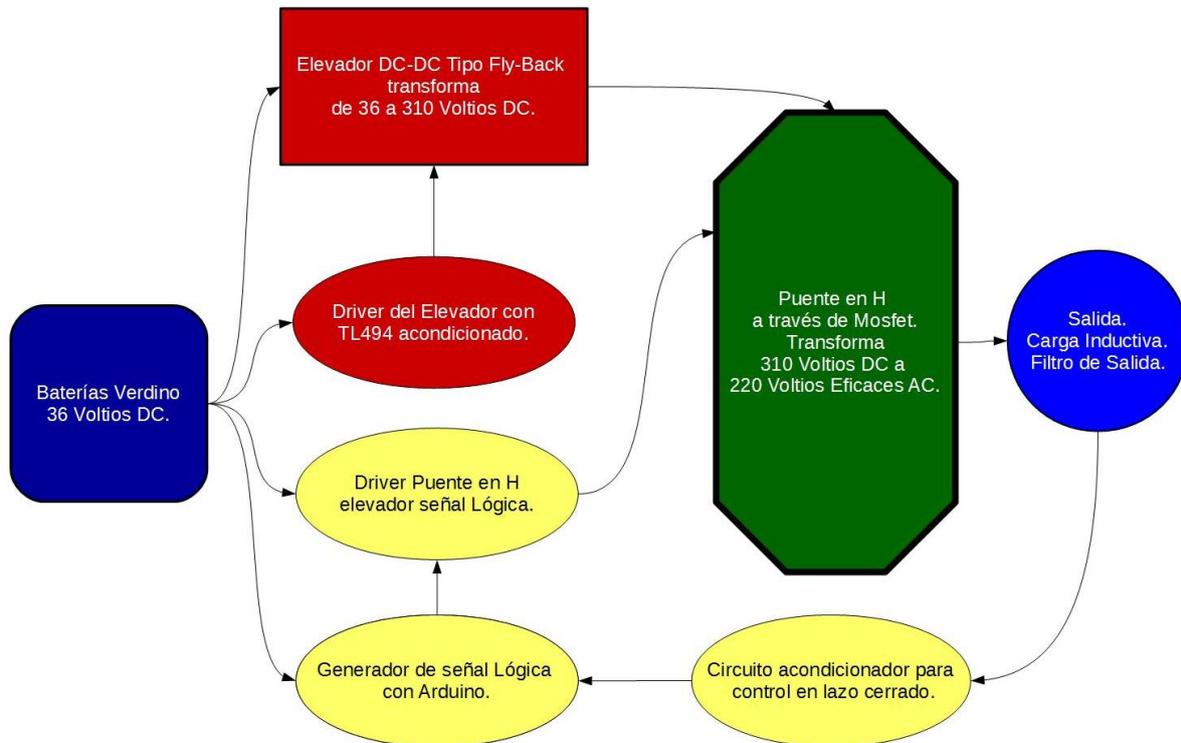


Figura 2: Esquema de la estructura del inversor

Primero, el sistema se ramifica en dos direcciones:

- Subir la tensión de continua de 36 a 310 Voltios. Para el cual se utiliza un convertor DC-DC con transformadores de tipo Fly-Back (se explicará en el apartado 2.1) y generar la señal lógica que lleva el interruptor de dicho convertor.
- Generar una señal digital de pulsos con el sistema Arduino de entre 0 y 5 voltios para luego elevarla en el driver, y que así, permita la conmutación de los interruptores del Puente en H.

Después ambas señales entran al Puente en H donde se generan 220 Voltios eficaces en la carga. En esta carga existirá un sistema de filtrado para mejorar la calidad de la señal de salida.

Por último, existe un pequeño circuito en la carga para la realimentación, donde se reduce la amplitud de la señal para que entre de nuevo en el Arduino.

1.5- Herramientas utilizadas para para el diseño del proyecto.

En este apartado se hace uso de definiciones sacadas de Wikipedia (2017).

1.5.1- Herramientas utilizadas en el montaje y medición.

1.5.1.1-Protoboard.

Una protoboard o placa de pruebas es una base de construcción para prototipos electrónicos. Es un tablero con orificios que se encuentran conectados eléctricamente de manera interna, habitualmente siguiendo patrones de líneas, en el cual se pueden insertar componentes electrónicos y cables para el armado y prototipado de circuitos electrónicos y sistemas similares.

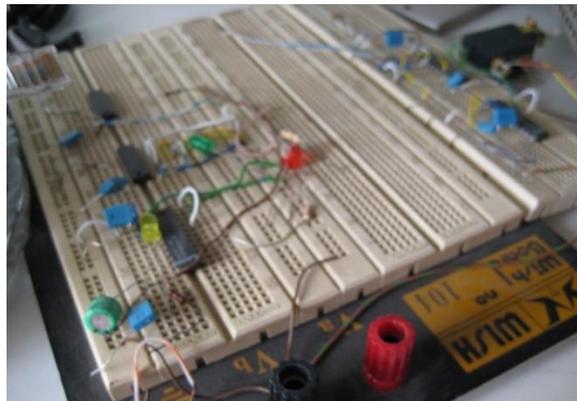


Figura 3: Protoboards con cableado y componentes de otro proyecto.

Atendiendo a esta definición, dentro de nuestro proyecto se hará uso de líneas verticales para los puntos de alimentación y tierra, y líneas horizontales para los componentes electrónicos. Se utilizarán varias protoboards en el desarrollo del proyecto.

1.5.1.2-Osciloscopio.

Un osciloscopio es un instrumento de visualización electrónico para la representación gráfica de señales eléctricas que pueden variar en tiempo.

El osciloscopio utilizado en el proyecto es digital, por lo que su funcionamiento es distinto, pero el resultado es el mismo y tiene un mayor número de ajustes para la visualización de la señal deseada.



Figura 4: Osciloscopio utilizado en el laboratorio.

1.5.1.3-Multímetro.

El multímetro, también denominado polímetro, o tester, es un instrumento eléctrico portátil para medir directamente magnitudes eléctricas activas, como corrientes y potenciales (tensiones), o pasivas, como resistencias, capacidades y otras.



Figura 5: Multímetro utilizado en el laboratorio.

1.5.1.4- Fuente de alimentación.

En electrónica, la fuente de alimentación es el dispositivo que convierte la corriente alterna, en una o varias corrientes continuas, que alimentan los distintos circuitos del aparato electrónico al que se conecta.



Figura 6: Fuente de alimentación del proyecto.

1.5.1.5 Generador de funciones.

El generador de funciones es una herramienta que permite la generación de varios tipos de señales alternas donde a través de los mandos se selecciona la forma de onda, la amplitud, su frecuencia y su valor de continua.



Figura 7: Generador de funciones utilizado en laboratorio.

1.5.2- Herramientas de software.¹

1.5.2.1- LTSpice IV.

LTSpice es un software gratuito que implementa un simulador de circuitos electrónicos, creado por el fabricante de semiconductores Linear Technology (LTC).

Este programa tiene como característica principal la capacidad de crear esquemas de circuitos y la de tener un visor de señales. Además, cuando se simula con este programa toma datos de todos los nodos, por lo tanto, con una sola simulación podemos conocer el estado de todo el circuito.

Para simular componentes electrónicos existe una base de modelos de los mismos, desde la que se pueden importar al esquema eléctrico.

1.5.2.2- Arduino IDE v1.8.2.

Arduino IDE es un programa diseñado para la programación de sistemas Arduino y apoyado fuertemente por su comunidad en el desarrollo y mejora del mismo. Este software se distingue de los demás del ámbito de programación de microcontroladores en que: es libre, gratuito, fácil de usar y tiene un amplio abanico de librerías. Arduino IDE utiliza lenguaje C y contiene un depurador y un compilador.

1.5.2.3- Libre office.

Libre office es una herramienta de ofimática de software libre y código abierto desarrollado por The Document Fundación como alternativa de OpenOffice. En este proyecto se ha utilizado para la escritura del mismo.

¹ Las siguientes definiciones son de elaboración propia.

2- CARACTERÍSTICAS DE DISEÑO

2.1- Tipos de inversores.

Existen muchos tipos de inversores, para el desarrollo de este proyecto se atenderá a denominarlos por su señal de salida. Según este criterio existen tres tipos:

- Onda cuadrada. Tiene un rendimiento bajo y puede generar ruido, siendo no apto para motores de inducción pero es el más barato de todos. Representado en morado en la gráfica inferior.
- Onda senoidal modificada. El ancho de la onda es modificado para acercarlo lo más posible a la onda senoidal. Aun así no llega a seguirla totalmente, sin embargo tiene un rendimiento mayor y menos armónicos que la de onda cuadrada. En el gráfico se representa de color azul.
- Onda senoidal pura. Son los más eficientes y permiten el uso de cargas sofisticadas. Son también los más caros y los que más electrónica llevan, suelen llevar microcontroladores para la generación de las señales lógicas. Se representa en negro en la gráfica².

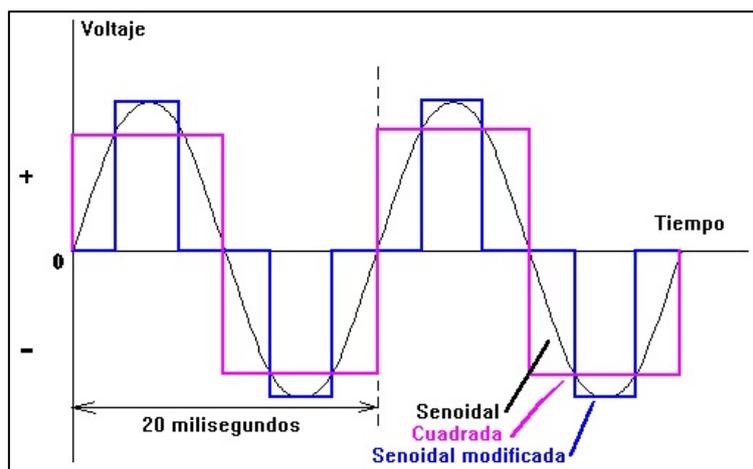


Figura 8: Esquema de los tipos de inversores según su salida.

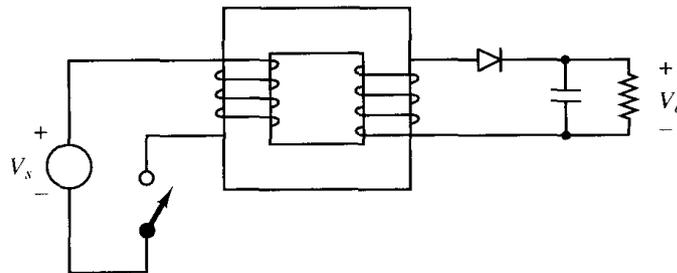
En este proyecto se busca una mejora respecto al inversor ya realizado por Néstor, en el cual se diseñó con una señal de salida por senoidal modificada.

² Información sacada del Artículo “Diferencias entre los distintos tipos de inversores” (2017)

2.2- Conversor DC-DC.

En este proyecto se usa un conversor DC-DC tipo Flyback que tiene la forma de la Figura 9 (a), donde se aísla la entrada de la salida a través de un transformador.

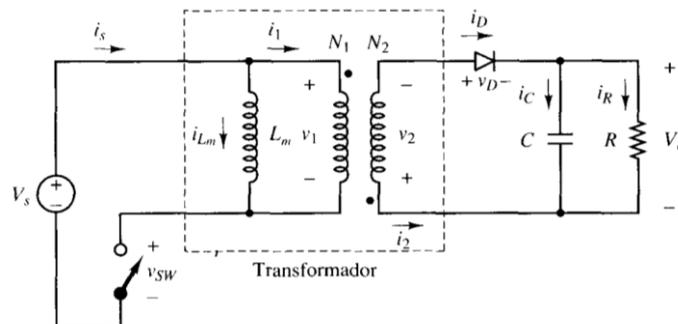
Para explicar este apartado se hará uso de información e imágenes obtenidas de: Hart, D. W. (2001). Electrónica de Potencia. Madrid: Pearson Education, S.A.



(a)

Figura 9 (a): Esquema de circuito Flyback.

En la Figura 9 (b) se muestra el circuito equivalente con un modelo de transformador que incluye la inductancia magnetizante.



(b)

Figura 9 (b): Modelo equivalente del circuito Flyback.

Cuando el interruptor está cerrado el transformador eleva la corriente y se invierte a la salida del mismo, el diodo se encuentra en inversa por lo que no conduce y el condensador está suministrando corriente a la carga.

Cuando el interruptor está abierto se transfiere la energía almacenada en la inductancia a la salida, polarizando el diodo en inversa y entregando corriente a la carga y al condensador de la salida.

La relación entre la tensión de entrada y salida es igual a la del convertidor DC-DC elevador-reductor, pero con un factor añadido de la relación de vueltas del transformador. Si se supone que el ciclo de trabajo es del 50 por ciento se obtiene que el voltaje de salida solo depende de la relación de vueltas de las bobinas de los transformadores.

$$V_o = V_s \frac{D}{1-D} \frac{N_2}{N_1}$$

Figura 10 (a):

Ecuación de salida para el convertor Flyback;
D es el ciclo de trabajo y N_x el número de vueltas de cada devanado.

El rizado en la señal de salida dependerá del paralelo entre la resistencia de la carga y el condensador, además de la frecuencia y el ciclo de trabajo del interruptor tal como se muestra en la figura 10 (b). Si se supone que el ciclo de trabajo la resistencia en la carga y la frecuencia son constantes, el rizado dependerá del condensador seleccionado donde cuanto mayor sea menor será el rizado y mejor será la señal de continua.

$$\frac{\Delta V_o}{V_o} = \frac{D}{RCf}$$

Figura 10 (b):

Ecuación del rizado de la salida; D es el ciclo de trabajo,
f la frecuencia, y RC son los valores de resistencia de salida y condensador.

2.3- Puente en H.

Un Puente en H es un circuito que se utiliza para convertir corriente continua en alterna. Existe la variante para media onda, pero en nuestro caso utilizaremos la de onda completa como se muestra en la figura 11(a). La salida será $+V_{cc}$ o $-V_{cc}$ dependiendo de que interruptores estén cerrados y para ello tenemos cuatro posibilidades.

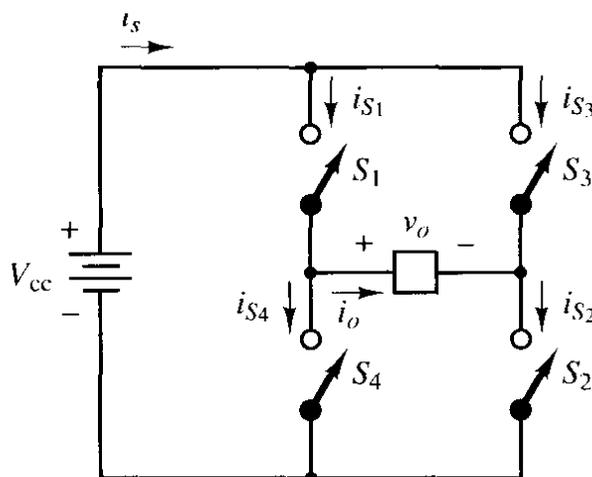
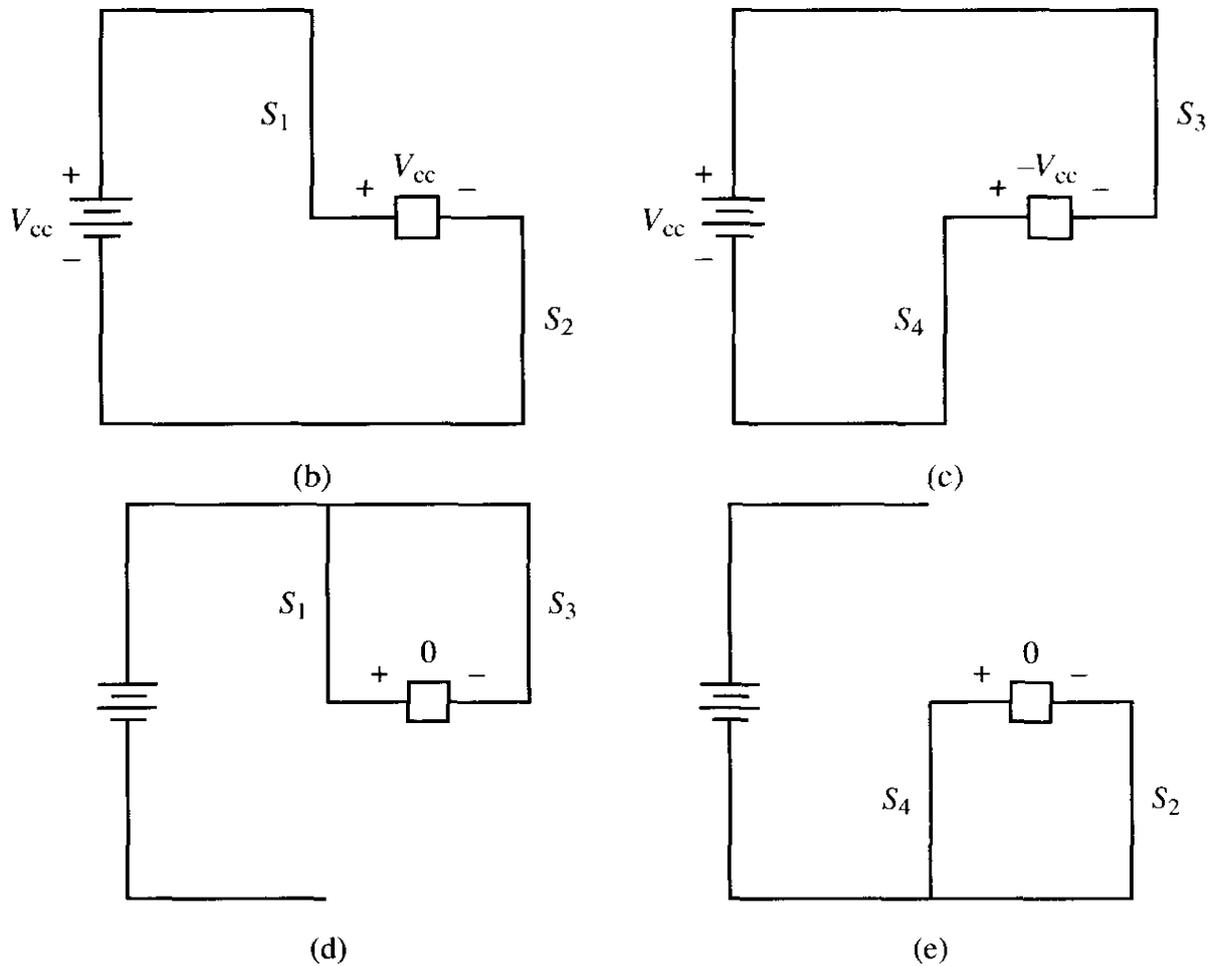


Figura 11 (a): Puente en H con interruptores ideales.

Que S1 y S2 estén cerrados, lo que genera que en la salida haya un valor de $+V_{cc}$. Que S3 y S4 están cerrados, dando lugar a que la corriente entre por el polo negativo generando un valor de $-V_{cc}$ en la carga.



Figuras 12:

(b) Puente en H con interruptores S1 y S2 cerrados.
(c) Con interruptores S3 y S4 cerrados.

(d) Con interruptores S1 y S3 conectados.
(e) Con interruptores S4 y S2 conectados.

Los casos en que S1 y S3, y S2 y S4 están encendidos simultáneamente se generará un voltaje de salida igual a 0. Nunca deben encenderse S1 y S4 o S3 y S2 a la vez ya que se produciría un cortocircuito en la fuente.

En la elección de los interruptores reales habrá que tener en cuenta su tiempo de apertura y cierre, para evitar que ocurran los estados no deseados incluso provocando que exista un tiempo muerto entre los interruptores S1- S2 y los S3-S4 para asegurarse de esto.

Se debe colocar un diodo de circulación libre en paralelo polarizado en inversa en cada interruptor para reducir los picos de corriente en conmutación debido a cargas inductivas, ya que permite el flujo de corriente durante los periodos de desconexión de los interruptores.

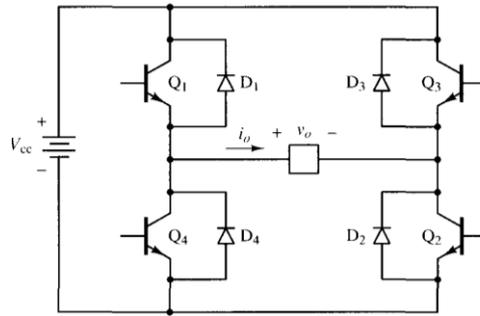


Figura 13: Puente en H utilizando transistores como interruptor, con diodos de libre circulación.

2.4- Driver del Puente en H.

Se precisará de una señal lógica para que los interruptores del Puente en H funcionen correctamente.

Como se enuncia en el apartado 2.1, existen tres tipos de señales lógicas.

Para este proyecto, se buscará que sea del tipo senosoidal pura, para lo que se necesitará una señal lógica más compleja donde los pulsos varíen su ancho dependiendo del tiempo.

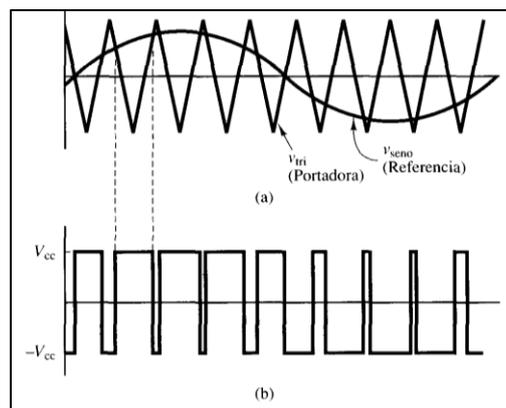


Figura 14:

- (a) Señal triangular portadora y señal senosoidal de referencia
- (b) Señal PWM modulada por anchura de pulsos.

El modelo para crear esta señal se realiza comparando una señal senosoidal, que tenga la frecuencia deseada en la señal de salida, con una señal triangular que definirá la cantidad de pulsos en cada periodo, provocando que la salida sea $+V_{cc}$ cuando la tensión de la señal senosoidal es mayor que la triangular y $-V_{cc}$ en caso contrario.

Para poder saber los valores de la amplitud de la señal de referencia y portadora, habrá que tener en cuenta el factor m_a (Índice de modulación de amplitud), si este es menor que 1 se cumple la equivalencia de la Figura 15 (b). Para conocer la relación entre las frecuencias portadoras y de referencia se utiliza el factor m_f (Índice de modulación de frecuencia). Se suele buscar un valor impar para este factor, ya que permite que la señal de salida de PWM tenga simetría impar y así se simplifique el cálculo (si fuera necesario) de la serie de Fourier.

$$m_a = \frac{V_{m, \text{referencia}}}{V_{m, \text{portadora}}} = \frac{V_{m, \text{seno}}}{V_{m, \text{tri}}}$$

(a)

$$V_1 = m_a V_{cc}$$

(b)

$$m_f = \frac{f_{\text{portadora}}}{f_{\text{referencia}}} = \frac{f_{\text{tri}}}{f_{\text{seno}}}$$

(c)

Figura 15:

- (a) Índice de modulación de amplitud.
- (b) Relación de m_a respecto a la salida.
- (c) Índice de modulación de frecuencia.

Para determinar la calidad de la señal de salida del Puente en H, se puede utilizar la expresión del valor de distorsión armónica total tanto para la tensión de salida como para la de corriente, que tienen expresiones equivalentes sustituyendo los voltajes por corrientes (en la ecuación de la figura 16) que sólo se puede aplicar de esta manera si no existen componentes de continua en la salida.

$$\text{DAT} = \frac{\sqrt{\sum_{n=2}^{\infty} (V_{n, \text{rms}})^2}}{V_{1, \text{rms}}} = \frac{\sqrt{V_{\text{rms}}^2 - V_{1, \text{rms}}^2}}{V_{1, \text{rms}}}$$

Figura 16: Expresión para el cálculo del factor DAT sin continua

Hay que tener en cuenta que la salida es una carga inductiva, por lo que habrá una carga RL que distorsionará la forma de onda de la salida, de manera que se colocará un filtro por condensador.

2.5- Variador de voltaje y frecuencia.

En este proyecto se va a implementar un variador de voltaje y frecuencia, que consiste en un sistema para controlar los parámetros del circuito durante el arranque. Se precisa la implementación de este sistema debido a la necesidad de evitar los efectos negativos del arranque de motores monofásicos, como picos de corriente que pueden provocar la sobrecarga de componentes del circuito y un consumo elevado durante esta etapa transitoria, que es muy superior al de la carga durante el estacionario.

En el puente en H una sobrecarga durante el arranque puede suponer la ruptura de uno de los mosfets, y esto a su vez, genera una sobretensión en el mosfet del mismo lado del puente, además de la posible destrucción del driver asociado a dicho lado.

Para ejemplificar esta situación se realiza una simulación de lo que ocurre cuando se arranca un sistema con una carga inductiva y lo que se puede observar es que sí se produce un pico de arranque, pero no muy significativo. Esto se debe a que la fase de la senoidal encaja con el arranque del sistema, y por lo tanto, es el caso más favorable durante el arranque.

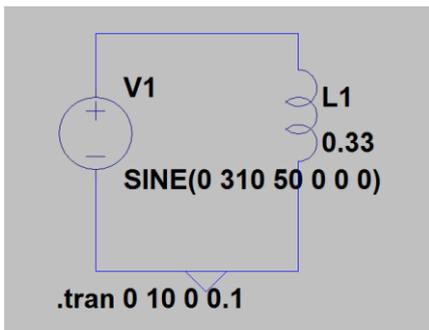


Figura 17: Esquema de carga inductiva simulando motor.

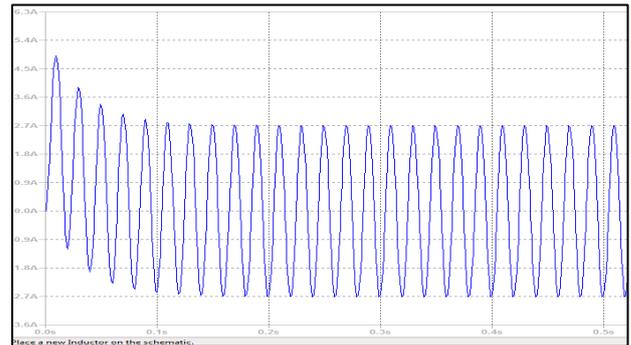


Figura 18: Gráfica de corriente de arranque en condiciones favorables.

Sin embargo, si se arranca con la fase de la senoidal desfasada 90° se obtiene el caso más desfavorable, donde el pico de arranque supera más de 15 veces su valor nominal.

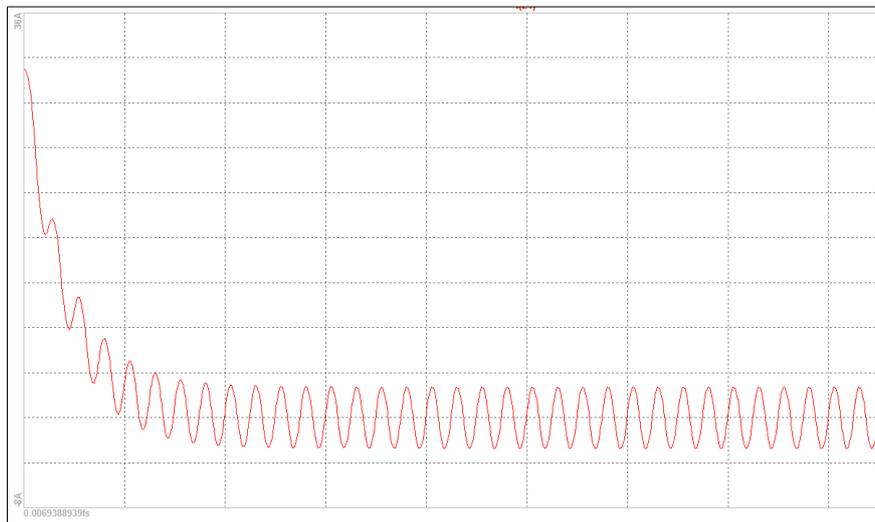


Figura 19: Gráfica de corriente de arranque en condiciones desfavorables.

El método que se plantea es el de diseñar un variador lineal que permita la energización progresiva del sistema evitando tanto el pico de arranque como el consumo excesivo. Sin embargo, esta idea implica que el sistema no tiene fuerza durante este periodo y por lo tanto genera un retraso en todo el sistema. En este caso, este hecho no influye ya que no es significativo un retraso a la hora de conectar el compresor para la eficiencia de los frenos.

3- PROGRAMACIÓN DEL ARDUINO

3.1- Características de Arduino.

3.1.1- Características de Arduino Uno.

Arduino Uno es una placa electrónica controlada por un microcontrolador Atmel328P.

Su alimentación puede variar de 7 a 12 voltios de continua, otros tipos de alimentaciones pueden dar como resultado salidas de menor tensión o la destrucción de los componentes por sobretensión.

Se puede conectar a través de un USB a un ordenador para su programación. Posee un cristal de cuarzo de 16MHz y un botón de reseteo rojo.

Tiene 14 pines digitales para input u output de los cuales, seis pueden ser usados para salidas PWM, y otros seis para entradas analógicas, todas ellas operan en un rango de 0 a 5 voltios. También tiene pines de voltajes DC de 5 y 3.3 voltios, además de dos pines para la puesta a tierra.

Dentro del microcontrolador Atmel328P hay que tener en cuenta que tiene un máximo de 32Kb de memoria (solo 2Kb de ellos son de SRAM en los que se puede guardar las variables del programa). Además, tiene 3 temporizadores: uno de 16 bits y dos de 8 bits cada uno.

En este proyecto se usa una placa electrónica equivalente a Arduino Uno, pero de una marca comercial distinta, la única diferencia es que precisa de la instalación de un driver manual para el enlace entre computadora y placa.

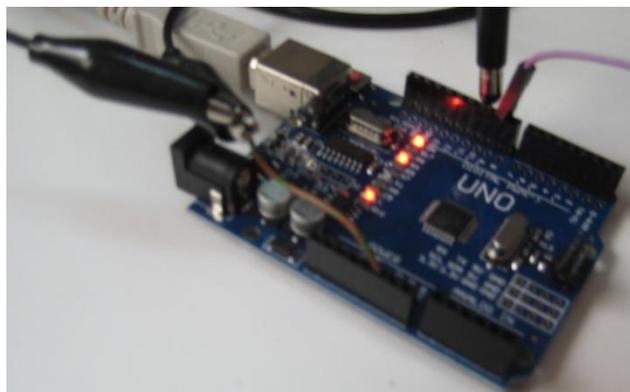


Figura 20: Placa Arduino utilizada en el proyecto.

3.1.2- Funcionamiento Arduino IDE.

Arduino IDE es un programa que funciona con lenguaje C, lo que permite el enlace de las órdenes del programador y el hardware de Arduino. Precisa de que se le indique: la placa electrónica a utilizar y el puerto al que se va conectar.

La programación tiene por defecto dos funciones que son las principales del sistema: la función Setup de variable void, que es una función que sólo se ejecuta una vez por lo que sirve para

inicializar el sistema; y la función Loop de tipo void, que se repite en bucle cada vez que termine la última línea del código por lo que su duración varía según el contenido.

Dentro de esta programación está permitido el uso de librerías externas y de funciones fuera de las mencionadas anteriormente.

3.2- Descripción del código.

El código utilizado es una modificación de un código libre extraído del usuario Terbytes de la página web <https://www.github.com/Terbytes/Arduino-Atmel-sPWM>.

La modificación se realizará dentro de la función Loop para generar el variador y el control en lazo cerrado, para ello también se modifica levemente el Setup y se añaden variables durante la declaración de estas. La parte modificada se ha colocado entre dobles barras de comentarios para su identificación.

El objetivo del código es conseguir que dos pines saquen una señal de modulación senosoidal por anchura de pulsos (SPWM). Al comparar una señal senosoidal con un temporizador se obtiene una forma de onda SPWM, activando intermitente cada pin de salida se alcanza la forma de onda deseada. El código utilizado es el siguiente:

```
#include <avr/io.h>
#include <avr/interrupt.h>
```

El código empieza introduciendo las librerías io.h y interrupt.h, que sirven para poder utilizar funciones que manipulen el microcontrolador interno directamente (permitiendo el uso de funciones como ISR y llamar a partes del controlador como TISK o DDRx), y para poder encender y apagar los interruptores globales respectivamente (funciones Cli para desactivar y Sei para encender).

A continuación, se definen las variables y se declaran las funciones adicionales.

El valor de frecuencia de la portadora se resolverá en el capítulo 4.2 de este proyecto.

```
#define LookupEntries (512)
//
#define referenceEntries (20)
//
static int microMHz = 16; // clock frequency in MHz
static int freq_ amp = 1024; // Sinusoidal frequency
static long int period; // Period of PWM in clock cycles. 1600 gives 10KHz.
static unsigned int lookUp[LookupEntries];
static char theTCCR1A = 0b10000010; //variable for TCCR1A
static unsigned long int phaseinc, switchFreq;
static double phaseincMult;
//
```

```

static int useFreq = 50;           // Frecuencia deseada de salida.
static int useSwitchFreq = 10000; // Frecuencia de la portadora.
static int useAmp = 100;          // Porcentaje de amplitud.
static float riseTime = 10000 ;   // Microsegundos dados hasta que el sistema se encuentre a plena carga.
static int resizeValue = 164;     // Valor de reescalado 1024*100/(310*1024/500)
static unsigned int reference[referenceEntries]; // Array de valores para el lazo cerrado
static int ampVal, freqVal, analogVal, reset , i , a , controlTime, resize2, resize;
static float currentTime;        // Variable para almacenar el tiempo de ejecución.
static float ampVal2;
static float freqVal2;
//
int setFreq(int freq);           //set in Hertz
int setSwitchFreq(int sfreq);   //set in Hertz
int setAmp(float _amp);         //set in % (0 - 100)
void makeLookUp(void);
void registerInit(void);

```

La **función registerInit** realiza una preparación en los registros de control de temporizadores y contadores. Con la configuración del TCCR1A se limpian los registros de las salidas del temporizador 1 (OCR1A y OCR1B), que corresponden a las salidas digitales de Arduino Uno de los pines 8 y 9, estos se configuran con la función DDRB como salidas que serán las que se enlacen a posteriori al circuito. Además, al colocar los bits del TCCR1A y TCCR1B con sus valores de WGM10:13 de la tabla 16-4 en el datasheet del Atmel328p con una configuración 1:1:1:0 se consigue que el sistema funcione como Fast-PWM.

Table 16-4. Waveform Generation Mode Bit Description⁽¹⁾ (Continued)

Mode	WGM13	WGM12 (CTC1)	WGM11 (PWM11)	WGM10 (PWM10)	Timer/Counter Mode of Operation	TOP	Update of OCR1x at	TOV1 Flag Set on
13	1	1	0	1	(Reserved)	-	-	-
14	1	1	1	0	Fast PWM	ICR1	BOTTOM	TOP
15	1	1	1	1	Fast PWM	OCR1A	BOTTOM	TOP

Figura 21: Tabla de generación de onda con el TCCRxA y TCCRxB

La **función TIMSK1** sirve para encender la función desbordar en el temporizador 1 (overflow) y que cuando este temporizador no pueda seguir contando, llame a la función TIMER1_OVF_vect .

```

void registerInit(void){
// Register initialization, see datasheet for more detail.
TCCR1A = theTCCR1A; // 0b10000010;
/*10 clear on match, set at BOTTOM for compA.
00 compB disconnected initially, toggled later to clear on match, set at BOTTOM.
00
10 WGM1 1:0 for waveform 15.
*/

```

```

TCCR1B = 0b00011001;
/*000
 11 WGM1 3:2 for waveform 15.
 001 no prescale on the counter.
*/
TIMSK1 = 0b00000001;
/*00000000
 1 TOV1 Flag interrupt enable.
*/
sei(); // Enable global interrupts.
// Set outputs pins.
DDRB = 0b00000110; // Set PB1 and PB2 as outputs.
pinMode(13, OUTPUT); // Set trigger pin to output
}

```

La **función makeLookUp** es una función que genera una tabla modulada en una señal senoidal, para ello se apaga los contadores e interruptores durante el proceso y se genera un bucle para rellenar un vector con dichos valores. Nótese que solo se genera la mitad de la señal senoidal, ya que cada pin de salida va a tomar el valor de media onda senoidal.

```

void makeLookUp(void){
  double temp;

  cli(); //disable global interrupts while lookup table is made
  TCCR1A = 0b00000010; //disconnect compare A and B while lookup table is generated

  for(int i = 0; i < LookupEntries; i++){ // Generating the look up table.
    temp = sin(i*M_PI/LookupEntries)*4096;
    lookUp[i] = (int)(temp+0.5); // Round to integer.
  }

  TCCR1A = theTCCR1A; // reconnect compare outputs
  sei(); //re-enable interrupts now that table has been made
}

```

La **función setAmp** verifica que la amplitud no sea menor que 0 ni mayor que 100 y reescala este valor respecto a 1024.

```

int setAmp(float _amp)
{
  if(_amp < 0 || _amp > 100){
    return 0;
  } else {
    amp = map(_amp, 0, 100, 0, 1024);
    return 1;
  }
}

```

La **función setSwitchFreq** va a realizar una comprobación donde la frecuencia de la portadora puede ser como máximo 20.000 Hercios, luego se calcula el periodo de la portadora dividiendo la velocidad del microcontrolador entre dicha frecuencia. Luego, se calcula el incremento de fase por cada periodo, para ello se usa un pequeño truco: se añade un factor de dos en potencia de 23 para obtener una palabra sin signo de 32 bites y que cuando se haga movimiento de bites los valores deseados queden los primeros, con esto se obtiene una menor pérdida de valores durante los saltos de la portadora al guardar (de mejor manera) los números decimales de la misma. Por último, se guarda el periodo en el registro ICR1 para asignar suficiente memoria al proceso.

```
int setSwitchFreq(int sfreq){
    double temp;

    if(sfreq <= 0 || sfreq > 20000){
        return 0;
    } else {
        switchFreq = sfreq;
        period = microMHz*1e6/sfreq;
        //sindevisions*decimalbits/1MHz =
        //1024*2^23/1e6 = 8,589.934592
        phaseincMult = (double) period*8589.934592/microMHz;
        phaseinc = (unsigned long int) phaseincMult*freq;
        ICR1 = period;
    }
}
```

La **función setFreq** comprueba que la frecuencia de la señal de salida no sea mayor de 1000 y calcula el incremento de fase de la misma manera que en la función anterior.

```
int setFreq(int _freq){
    if(_freq < 0 || _freq > 1000){ // returns -1 if the frequency value is invalid
        return 0;
    } else {
        freq = _freq;
        phaseinc = (unsigned long int) phaseincMult*_freq;
        return 1;
    }
}
```

Se modifica la función por defecto del desborde del Timer1 accediendo al ISR. A continuación, se busca que cada fase sea la suma de las fases anteriores. Luego, se comprueba si el contador “phase” se ha desbordado, si eso ocurre se resetea encendiendo y apagando el registro TCCR1A para limpiar el buffer del mismo. Al hacerlo en un nuevo ciclo, se consigue que encaje con el inicio de los valores de la tabla. Al final, se carga el valor deseado en los registros de salida, que en este caso consiste en mover bits dependiendo de la variable para que el resultado sea una palabra de nueve bits.

```

ISR(TIMER1_OVF_vect){
    static unsigned long int phase, lastphase;
    static char delay1, trig = LOW;

    phase += phaseinc;

    if(delay1 == 1){
        theTCCR1A ^= 0b10100000;// Toggle connect and disconnect of compare output A and B.
        TCCR1A = theTCCR1A;
        delay1 = 0;
    }
    else if((phase>>31 != lastphase>>31) && !(phase>>31)){
        delay1++;
        trig = !trig;
        digitalWrite(13,trig);
    }

    lastphase = phase;
    OCR1A = OCR1B = ((lookUp[phase >> 23]*period) >> 12)*amp >> 10;
}

```

La **función Setup** ejecuta las funciones por primera vez y coloca el pin analógico A0 como input para controlar su corriente máxima.

```

void setup(){
    Serial.begin(9600);
    makeLookUp();
    setSwitchFreq(useSwitchFreq);
    setAmp(useAmp);
    registerInit();
    //
    pinMode(A0, INPUT); //Añadido para asegurarnos que no se quema el pin por sobrecorriente.
    //
}

```

La **función Loop** es la encargada de crear el variador y el sistema en lazo cerrado. Para ello, se divide en tres secciones:

La primera sección es la encargada del encendido, donde la amplitud y la frecuencia de la señal de salida escalan entre un factor que varía linealmente entre 0 y 100 dependiendo del tiempo de ejecución. Durante este tiempo, se hace un control en el que se recoge el valor del pin analógico y se reescala a un valor de 164 (donde 100 es el valor máximo deseado de la señal) y si este sobrepasa un 5% el valor deseado, se reajusta la salida. Como denota este código, solo se controla que no sobrepase el máximo durante este periodo.

```

void loop(){
//
//Encendido del sistema, con lazo cerrado, solo control de exceso de tensión.

currentTime=millis();
if (currentTime <= riseTime){
freqVal2 = (float) useFreq;
freqVal =(int) currentTime * freqVal2 / riseTime;
ampVal2= (float) useAmp;
ampVal=(int) currentTime * ampVal2 / riseTime;
setFreq(freqVal);
setAmp(ampVal);
analogVal=analogRead(A0);
resize=map(analogVal,0, 1024 , 0 ,resizeValue );
if (ampVal*1.05 < resize){
resize2=2*ampVal-resize;
setAmp(resize2);
}
}
// Punto de cruce
if (currentTime < riseTime+100 && currentTime > riseTime){ // Intervalo de paso entre encendido y estacionario.
setSwitchFreq(useSwitchFreq); // Llamada a la funciones para control.
setFreq(useFreq);
setAmp(useAmp);
reset = 0; // Iniciación variable de set y reset
a = (int) 0; // Iniciación variable contador
}

```

La segunda sección es un punto intermedio, donde se fuerza que se ejecuten las funciones de amplitud y frecuencia con el valor deseado, para que el sistema llegue a alcanzar ese punto al final del encendido. Hay que tener en cuenta, que con este código el ciclo tarda más de un milisegundo, por lo que se debe dar margen a la condición que permite la entrada a esta sección.

Por último, la tercera sección es aquella en la que se almacenan los datos de un ciclo entero para conocer su valor máximo y así poder comparar este valor con el deseado para rectificar. Cuando se hayan almacenado 20 valores, o hayan pasado 20 milisegundos, se reinicia el vector y se actualiza el tiempo máximo para el siguiente ciclo. Como característica de diseño, este sistema no se puede aumentar su voltaje más del 100% de su capacidad, por lo que si el valor de salida es menor, el sistema no podrá compensarlo (este hecho se refleja en la función setAmp, donde si se introduce un valor mayor que 100, no hará nada).

```

if (currentTime >= riseTime+100){
if ( reset = 0){
memset(reference,0, sizeof(reference));
reset = 1;
controlTime = currentTime + 20;
}
if (reset = 1){
analogVal=analogRead(A0);
reference[a] = analogVal;
a=a+1;
if ( currentTime > controlTime || a <= 20){
ampVal = reference[0];
for( i=0; i<=referenceEntries; i++){
if ( reference[i] > ampVal){
ampVal= reference[i];
}
}
ampVal = map (ampVal, 0 , 1024, 0, resizeValue);
if (ampVal*1.05 < useAmp || ampVal*1.05 > useAmp){
resize2= 2*useAmp-resize;
setAmp(resize2);
}
}
reset =0 ;
a = 0;
}
}
//
}

```

// If para el estacionario.
// Reseteo interno del ciclo.
// Limpieza del array.
// Tiempo de control de 20 ms, periodo de la señal de salida.
// Ciclo de adquisición de datos.
// Leemos el valor del pin analógico.
// Introducimos el valor en el array.
// Aumentamos el contador para el siguiente ciclo.
// Condición del ciclo.
// Asignamos el primer valor del vector.
// Bucle para hayar el valor más alto del array.
// Comparación con el dato anterior más alto.
// Asignación del nuevo valor más alto.
// Reescalado del valor de analógico al porcentaje de amplitud.
// Comparación con un margen del 5%.
// Generación de nuevo valor de amplitud.

La forma en la que está hecho este programa permite cambiar la frecuencia de la carga y de la portadora con mucha facilidad. También genera la posibilidad de que si se coloca el sistema con más voltaje del deseado, regulando el porcentaje de la amplitud, se tendrá un control al exceso de tensión y a la falta del mismo.

3.3- Comprobación de funcionamiento.



Figura 22: Señal SPWM de la salida del Arduino Uno, la señal tiembla.

Se realizó una prueba de conexión conectando el sistema Arduino Uno (ya programado) a un osciloscopio, mientras era alimentado por una fuente de alimentación de cinco voltios.

El resultado obtenido es muy bueno, obteniendo una señal que varía su ancho modulada en una onda senosoidal y donde cada pin de salida funciona con exclusividad respecto al otro.

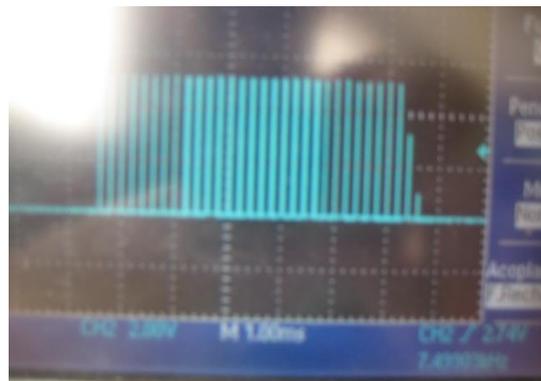


Figura 23: Ampliación de un pulso de la señal SPWM

La comprobación del variador y el sistema en lazo cerrado es complicada de visualizar, ya que es dinámico, por lo que se resolverá en el siguiente apartado.

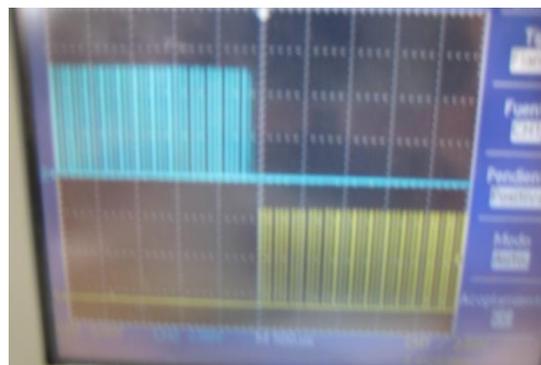


Figura 24: Ampliación de cómo encajan las dos señales SPWM, con trigger correcto.

4- IMPLEMENTACIÓN

4.1- Implementación de Elevador DC-DC.

En este apartado, se hará referencia al trabajo realizado por el compañero Néstor Gómez Tejera.

Como ya se ha mencionado antes, esta parte fue realizada previamente por Néstor Gómez Tejera. Todos los cálculos ya han sido realizados en su trabajo, por lo tanto, se partirá de la premisa de que éste apartado ya ha sido resuelto. Aun así, se mostrará cómo ha sido realizado para que se comprenda mejor el presente proyecto.

Es necesario obtener 220 voltios de alterna en la carga, por lo que con la ecuación del valor eficaz de una señal senoidal se puede hallar el voltaje pico necesario.

$$\boxed{V_{ef} = V_{rms} = \sqrt{\frac{1}{T} \int_0^T v^2(t) dt}} \longrightarrow \boxed{V_{rms} = \frac{V_m}{\sqrt{2}}}$$

Figura 25: Ecuaciones del valor eficaz y el valor eficaz en una senoidal: Hart, D. W., (2001)

Donde tras resolver la ecuación anterior ecuación con el valor deseado se obtiene un valor de 310 voltios aproximadamente.

$$\boxed{\begin{aligned} V_m &= V_{rms} * \sqrt{2} \\ V_m &= 220 * \sqrt{2} = 311,13V \end{aligned}}$$

En la Figura 26 se muestra el circuito utilizado en el proyecto del compañero, se utiliza el componente TL404 para controlar los mosfets M1, M2, M3 y M4 que son los encargados de generar una señal discontinua capaz de inducir corriente en los transformadores y además mantener una señal constante controlada a través de una realimentación regulada por un partidor de tensión con resistencias.

La relación de los transformadores necesaria en conjunto ha de tomar el siguiente valor según la ecuación de la figura 10 (a):

$$\begin{aligned} V_o &= V_s * \left(\frac{N_2}{N_1}\right) \rightarrow \frac{V_o}{V_s} = \frac{N_2}{N_1} \\ \frac{N_2}{N_1} &= \frac{310}{36} = 8,61 \end{aligned}$$

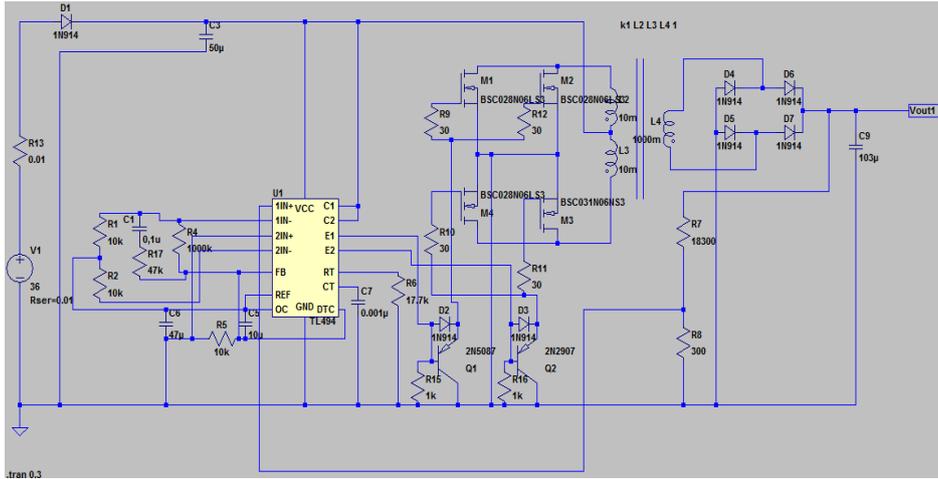


Figura 26: Circuito Flyback con controlador TL494 sacado (recuperado de Gómez. N., 2016)

El montaje de dicho circuito se muestra en la figura 27, donde se le conecta un transformador para comprobar que funciona correctamente.

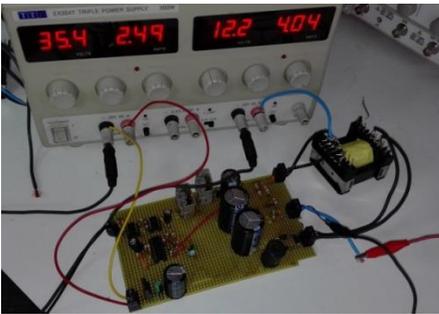


Figura 27: Montaje de circuito Flyback.

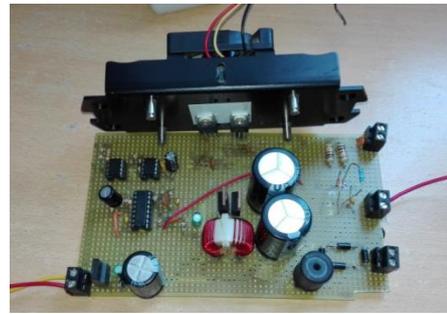


Figura 28: Circuito Flyback con disipador conectado a los Mosfet.

En las Figuras 28 y 29 se muestra la simulación y montaje de conexión de los tres transformadores necesarios para llegar hasta los 310 voltios de continua. Hay que tener en cuenta que el transformador intermedio está conectado en inversa.

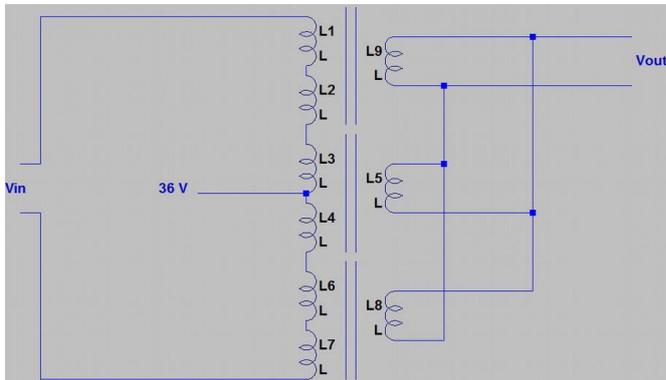


Figura 29: Esquema de conexión de los tres transformadores.

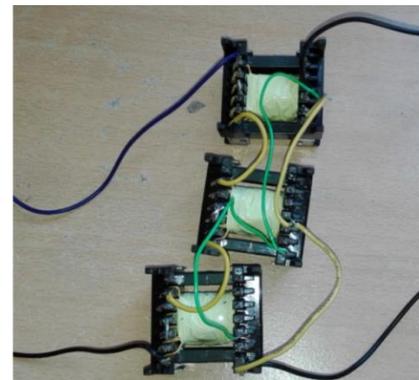


Figura 30: Montaje de los tres transformadores.

Debido a que los Mosfet se sobrecalentaban al conectar el sistema se colocó un disipador como muestra la figura 30.

4.2- Implementación de Driver del Puente en H.

4.2.1- Frecuencia de la portadora en la Spwm

En este apartado, haciendo referencia a información sacada de Hart, D. W. (2001), se procede al cálculo de la frecuencia necesaria para señal de la portadora en la Spwm según lo visto en el apartado 2.4 de este proyecto.

Primero, se calcula el factor m_a teniendo en cuenta que el voltaje de salida deseado está en valor eficaz y hay que pasarlo a voltaje pico.

$$m_a = \frac{V_s}{V_{cc}} = \frac{220 * \sqrt{2}}{310} = 1.004$$

Luego se calcula la amplitud de la intensidad de salida suponiendo un caso altamente desfavorable con una carga de 10 omhios y 5 miliHendrios

$$I_s = \frac{V_s}{Z_s} = \frac{220\sqrt{2}}{\sqrt{10^2 + (2 * \pi * 50 * 0.05)^2}} = 16.7084A$$

El valor rms de los armónicos de corriente tiene un límite impuesto por el factor DAT. Si se toma como extremo un DAT del 1% se tiene que:

$$\sqrt{\sum_{n=2}^{\infty} (I_n \cdot ef)^2} \leq DAT * I_1 \cdot ef = 0.01 * \left(\frac{16.7084}{\sqrt{2}}\right) = 0.1181A$$

La amplitud del armónico de corriente a la frecuencia de la portadora será aproximadamente:

$$I_{mf} < (0.01) * (16.7084) = 0.1671$$

Para calcular el armónico de tensión hay que ir a la tabla de la figura 31, donde si se busca para $n=mf$ y $m_a=1$, se tiene un factor 0.6 que aplicado a la tensión se obtiene:

$$V_{mf} = 0.6 * V_{cc} = 310 * 0.6 = 186V$$

	$m_a = 1$	0,9	0,8	0,7	0,6	0,5	0,4	0,3	0,2	0,1
$n = 1$	1,00	0,90	0,80	0,70	0,60	0,50	0,40	0,30	0,20	0,10
$n = m_f$	0,60	0,71	0,82	0,92	1,01	1,08	1,15	1,20	1,24	1,27
$n = m_f \pm 2$	0,32	0,27	0,22	0,17	0,13	0,09	0,06	0,03	0,02	0,00

Figura 31: Tabla de coeficientes de Fourier normalizados V_d/V_{cc} para PWM bipolar.³

La impedancia mínima en la frecuencia de la portadora se calcula dividiendo los dos valores anteriores.

$$Z_{mf} = \frac{V_{mf}}{I_{mf}} = \frac{186}{0.1181} = 1574.94\Omega$$

Si se presume que la inductancia es mucho más grande la resistencia de carga a la frecuencia de la portadora, se obtiene una reactancia totalmente inductiva:

$$Z_{mf} \approx \omega * L = m_f * \omega_0 * L$$

Para que la impedancia de la carga sea mayor que el valor mínimo:

$$m_f * \omega_0 * L > 1574.94\Omega$$

De aquí se calcula el factor m_f y se despeja la frecuencia de la portadora.

$$m_f > \frac{1574.94}{2 * \pi * f * L} = \frac{1574.94}{2 * \pi * 50 * 0.05} = 100.26$$

Al elegir el siguiente valor impar para m_f , se obtiene que:

$$f_{tri} = m_f * f_{ref} = 101 * 50 = 5050Hz$$

Con una señal de portadora con la frecuencia anterior se conseguiría el objetivo de diseño de tener un DAT adecuado. En la práctica, se utilizó un valor de 10000Hz para la señal portadora que casi duplica el mínimo, por lo que se tiene una señal con la mínima distorsión debido a la forma de la señal.

³ Disponible en: Hart, D. W. (2001). Electrónica de Potencia. Madrid: Pearson Education, S.A.

4.2.2- Simulación de Arduino Uno

Debido a que no existe ninguna manera de tener un modelo de Arduino Uno en el programa de simulación utilizado (LTSpice), se ha sustituido por un circuito que genera una salida similar a la que se obtiene de Arduino. La solución consiste en utilizar el esquema teórico, para ello, se comparan dos señales senosoidales desfasadas 90 grados entre sí, con una señal triangular a través de un Amplificador Operacional como el TL082 alimentado positiva y negativamente al mismo voltaje y luego se le restan las señales resultantes; como no se desea una señal de salida con valores negativos, se pondrá a tierra la alimentación negativa de este segundo Amplificador Operacional. Si se realizan estos por duplicado cambiando el orden de la resta se obtienen dos señales Spwm entre 0 y 5 voltios, tal y como se muestra en la figura 33.

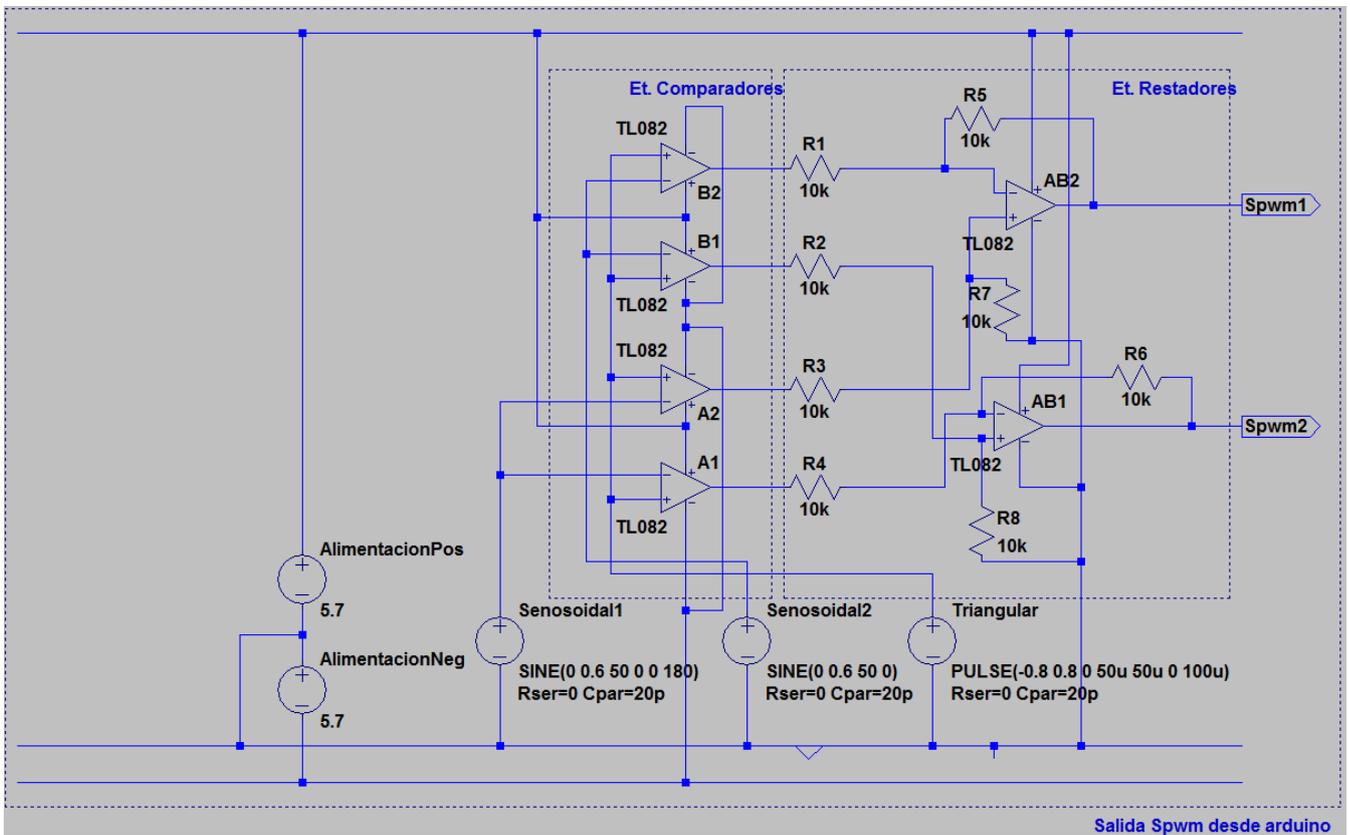


Figura 32: Esquema de simulación de salida Spwm en LTSpice

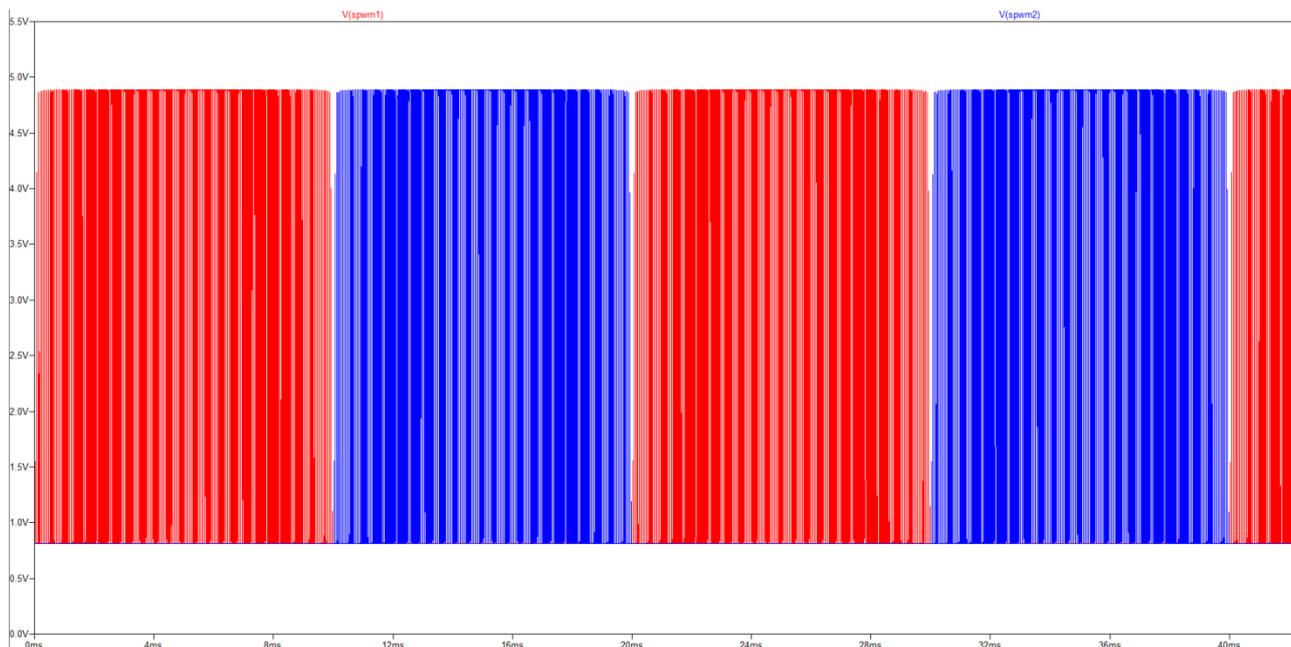


Figura 33: Simulación de la salida del Arduino en LTSpice.

4.2.3- Elevador de la señal Spwm

En este tipo de circuitos lo más correcto es empezar por un circuito de baja tensión e ir aumentando la tensión poco a poco con el fin de evitar daños materiales graves. Por ello, los circuitos se simularan en baja tensión, pero en la práctica se colocarán los componentes de alta tensión y se irá aumentado progresivamente la tensión de alimentación.

Para poder alimentar el Puente en H se necesita una señal lógica con una tensión mayor que la salida de Arduino Uno (entre 0 y 5 Voltios) que esté entre 10 y 20 Voltios. Hay que tener en cuenta que tal y como está diseñado el puente para que exista una diferencia de potencial entre los mosfet superiores y los inferiores, ya que estos conmutan con la misma señal, es necesario que los mosfet superiores tengan mayor el doble de tensión en su Puerta.

La primera hipótesis consiste en un sistema elevador que utiliza transistores para elevar la corriente y lleva dos mosfets adicionales para elevar por segunda la tensión en los mosfets superiores.

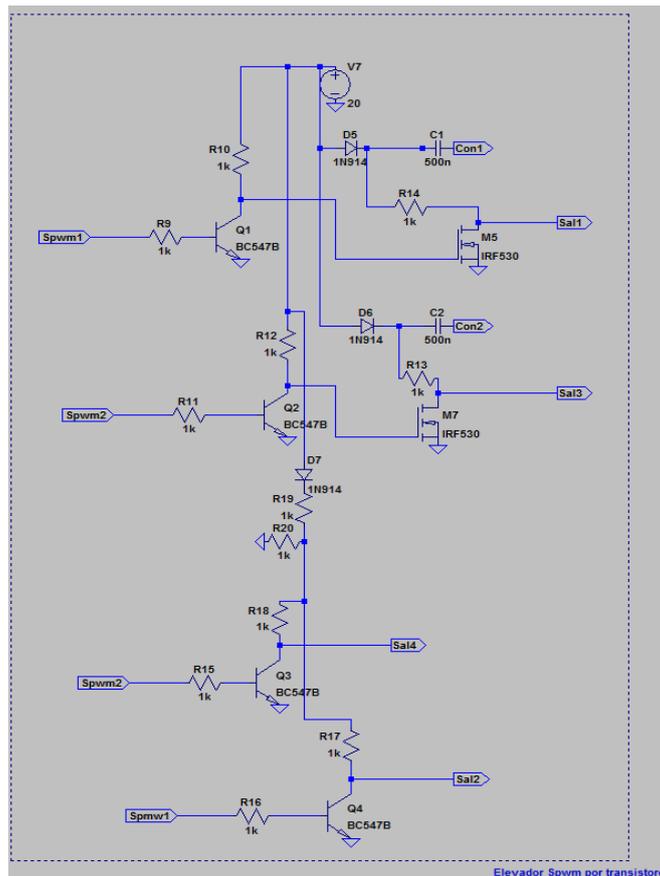


Figura 34: Esquema de elevador por transistores en LTSpice.

El esquema es análogo al utilizado por Néstor en su proyecto y se descubrió que los mosfet dejaban de funcionar correctamente debido a problemas de conmutación y sobrecalentamiento por disipación excesiva de potencia, por lo que en este proyecto se sustituye ese sistema por uno en donde la pieza central es un componente diseñado para este tipo de funciones.

Para baja tensión se usa el componente LT1162, que según su datasheet, es un driver de medio puente de hasta 60 voltios, que como se puede observar, no alcanza los 310 voltios que exige nuestro circuito, pero componentes con esos rangos no tienen modelo para los simuladores Spice. Por lo que, se simulará en baja tensión, y como el acondicionamiento es equivalente en la práctica solo se tendrá que sustituir uno por otro. El esquema del circuito utiliza un dispositivo por cada mitad del puente con las señales lógicas intercambiadas entre sí.

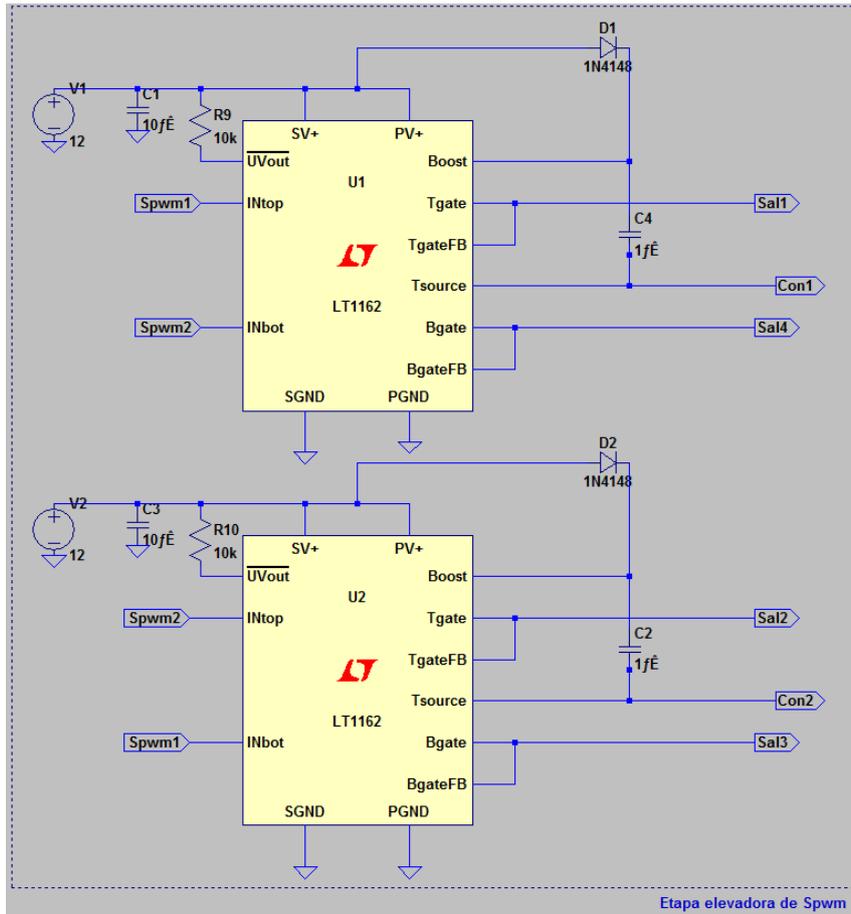


Figura 35: Esquema etapa elevadora de Spwm en LTSpice.

Los condensadores de la figura 35 están elegidos con un orden de magnitud de diferencia, que es lo que se especifica en el datasheet; nótese que la salida Tsource es la que va a generar que la salida Tgate sea superior a la Bgate debido que proviene de la carga y permite generar una diferencia de tensión entre la parte superior e inferior del puente.

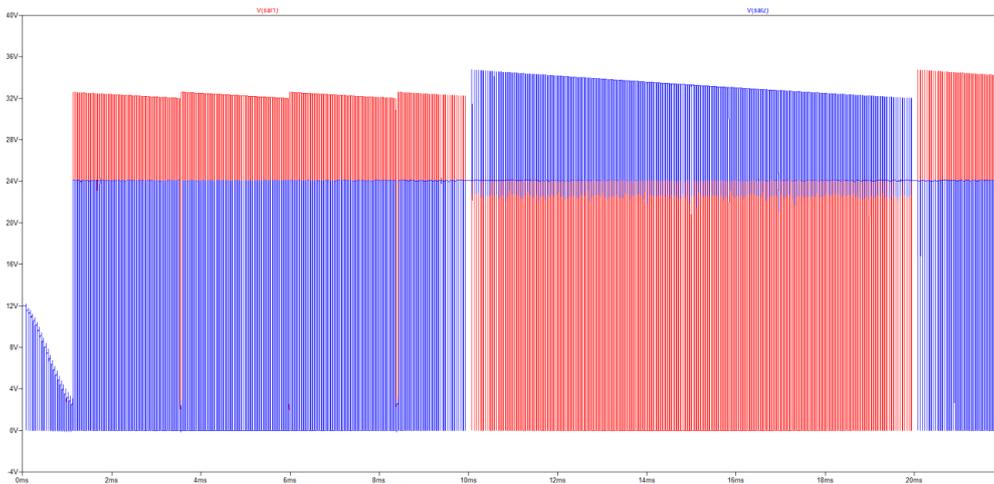


Figura 36: Voltaje de entrada a la puerta de los mosfets superiores.

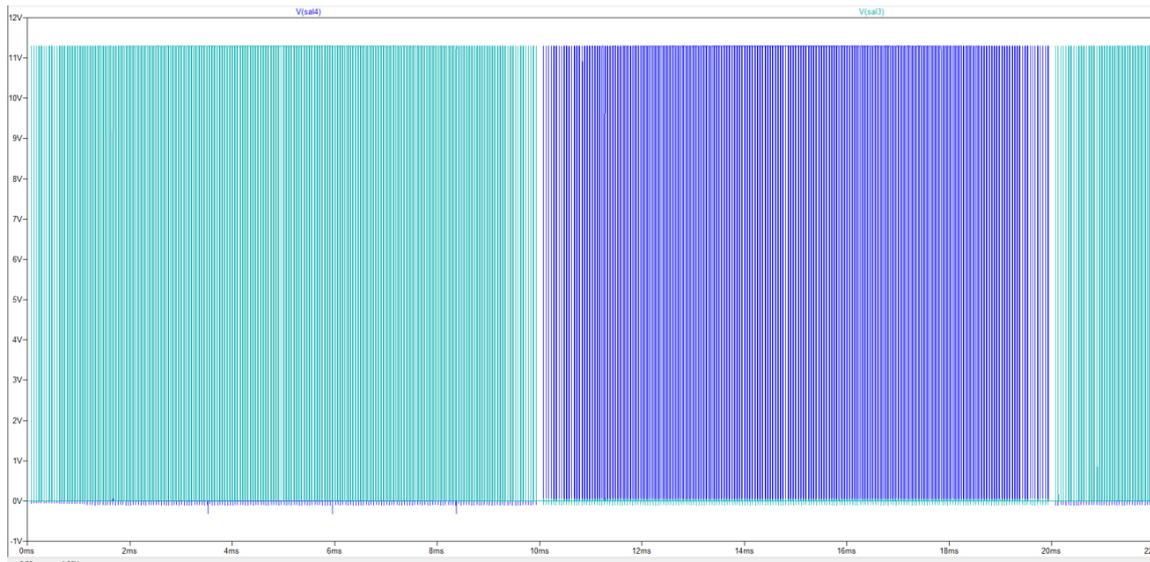


Figura 37: Voltaje de entrada a la puerta de los mosfets inferiores.

A la hora de hacer el montaje, se tuvo en cuenta que el componente L6387R no lleva el diodo en el acondicionamiento, sino que lo lleva ya integrado según su datasheet, por lo que fue removido.

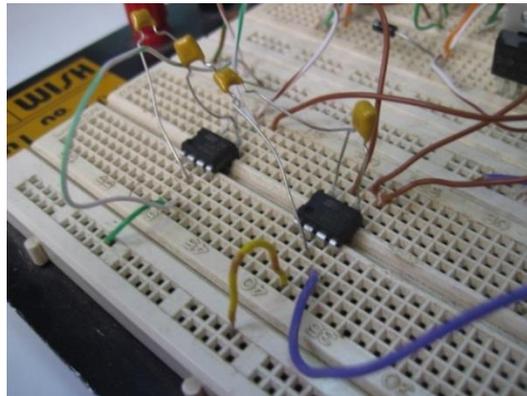


Figura 38: Fotografía del montaje del elevador de Spwm.

El montaje del circuito que se puede ver en la figura 38 y la salida de este circuito en la 39. Como se puede observar este circuito eleva la señal y le da a la parte superior una forma distinta a la inferior a la espera de que se conecte la carga para elevarse como se observa en la figura 40.



Figura 39: Fotografía de la salida del driver en el osciloscopio (Amarillo salida Bgate, azul salida Tgate)

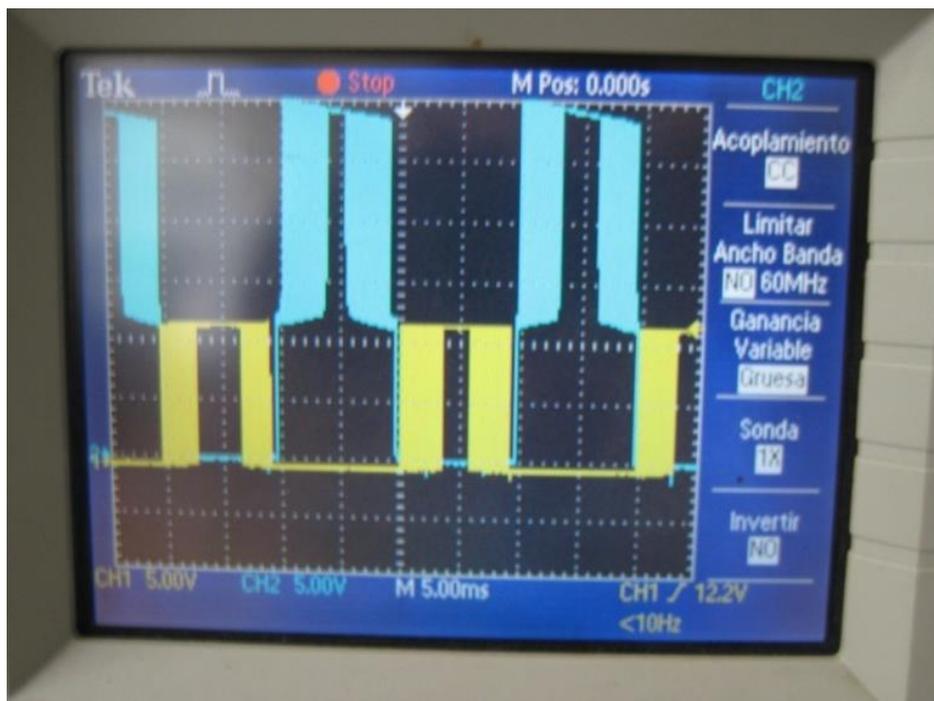


Figura 40: Fotografía de las salidas del driver con carga conectada. (Amarillo salida Bgate, azul salida Tgate)

4.3- Implementación de Puente en H.

Para la implementación del Puente en H solo se podrá simular el sistema en baja tensión debido a que el driver no funciona a alta tensión. En la simulación se utilizan dos tipos de carga: resistiva e inductiva, en esta última se añadirá un condensador en paralelo como filtro. Se mostrarán la tensión de salida y las potencias de consumo más relevantes, así como, los valores promedio y eficaces de las señales.

El esquema utilizado es el siguiente:

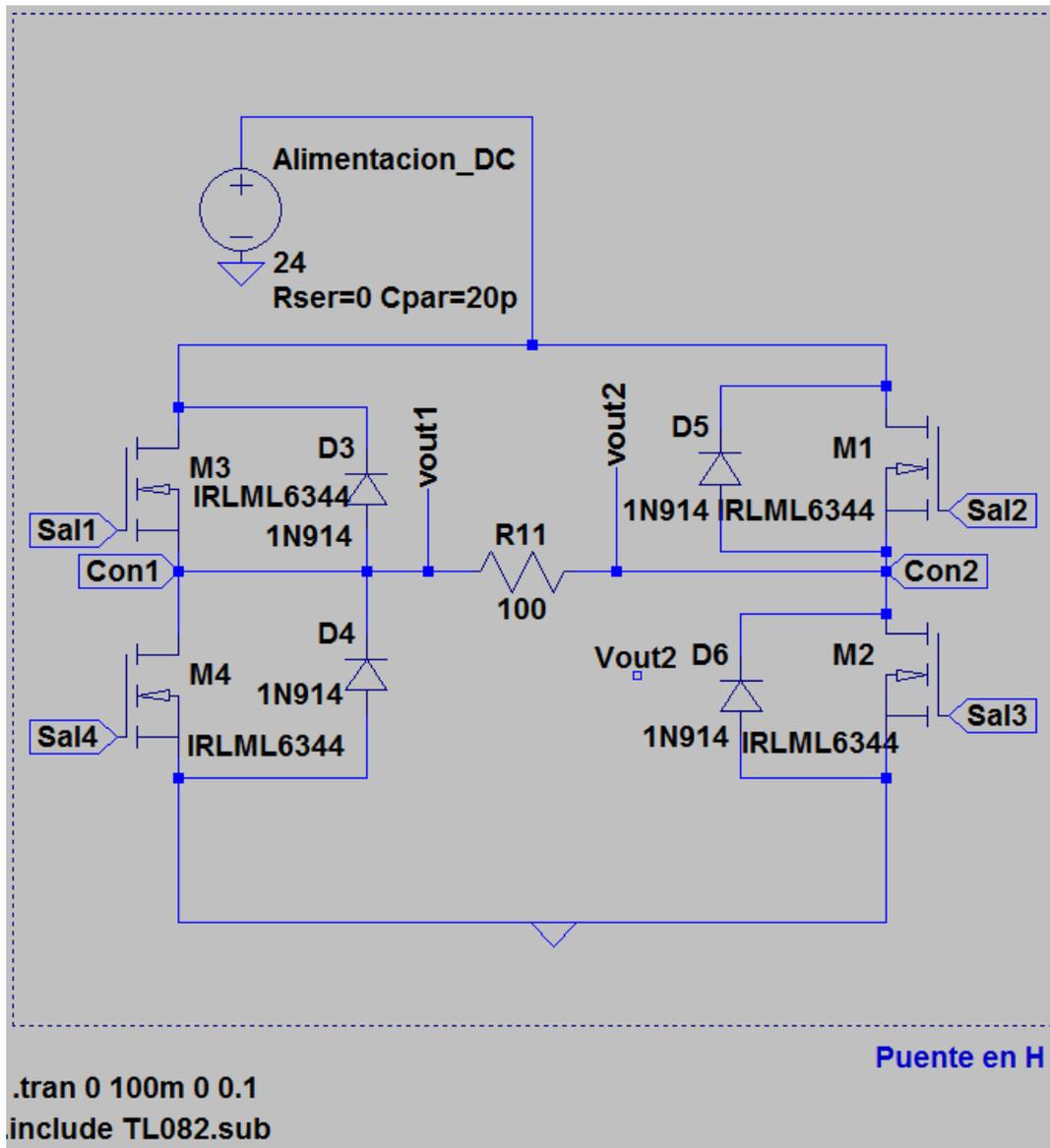


Figura 41: Esquema de la simulación del puente en H con carga resistiva.

En las figuras 42 y 43 se visualiza la señal de salida y sus valores promedio y eficaces. Se obtiene un voltaje pico de casi 24 voltios, un valor promedio casi nulo (-0.3mV) y un resultado eficaz de 16.37V muy cercano al valor teórico (16.97V).

A través de esta gráfica se puede comprobar que es lo que realiza el Puente en H: toma la señal de las entradas lógicas y la eleva a la tensión de pico de la alimentación tal como se explicó en el apartado 2.3.

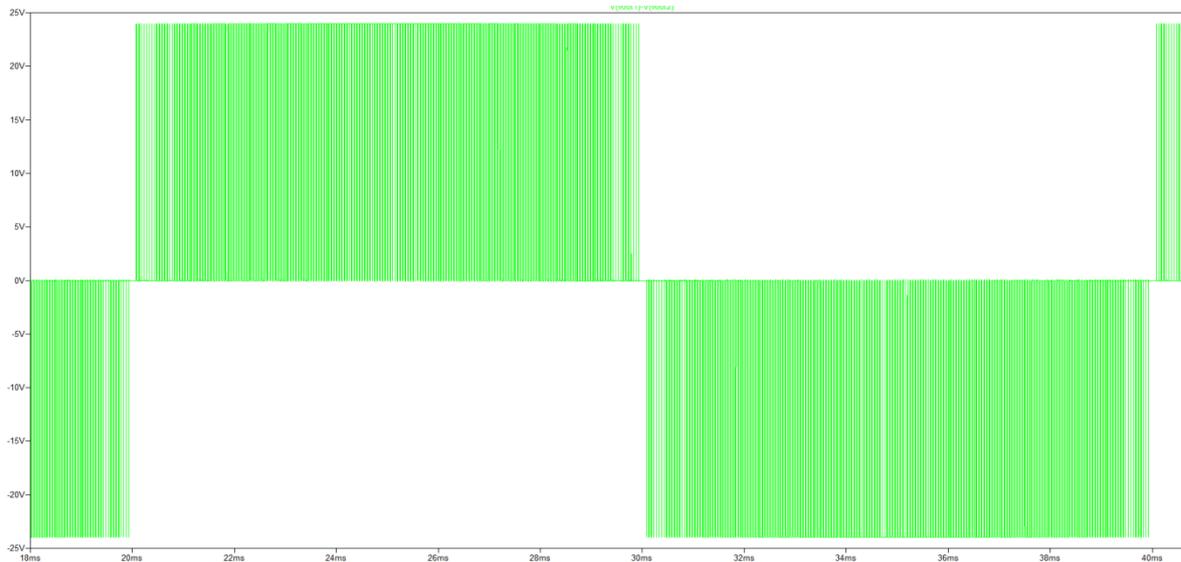


Figura 42: Salida del puente en H, simulación carga resistiva.

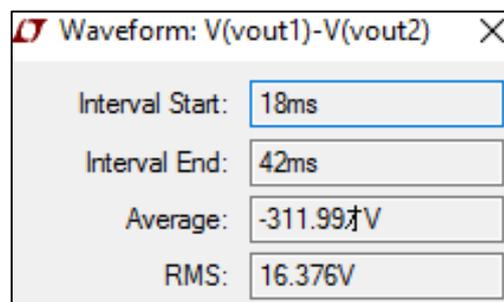


Figura 43: Valores promedio y eficaces de la señal de salida con carga resistiva.

A continuación se muestran las gráficas, valores promedio e integrales de la potencia de los elementos más susceptibles. Si se miran los valores de las integrales de la salida con respecto a la alimentación se puede observar que casi el 100% de la potencia es consumida en la carga.

También se puede apreciar que los mosfet tanto superiores como inferiores consumen prácticamente lo mismo, y que, la potencia disipada en ellos es casi nula.

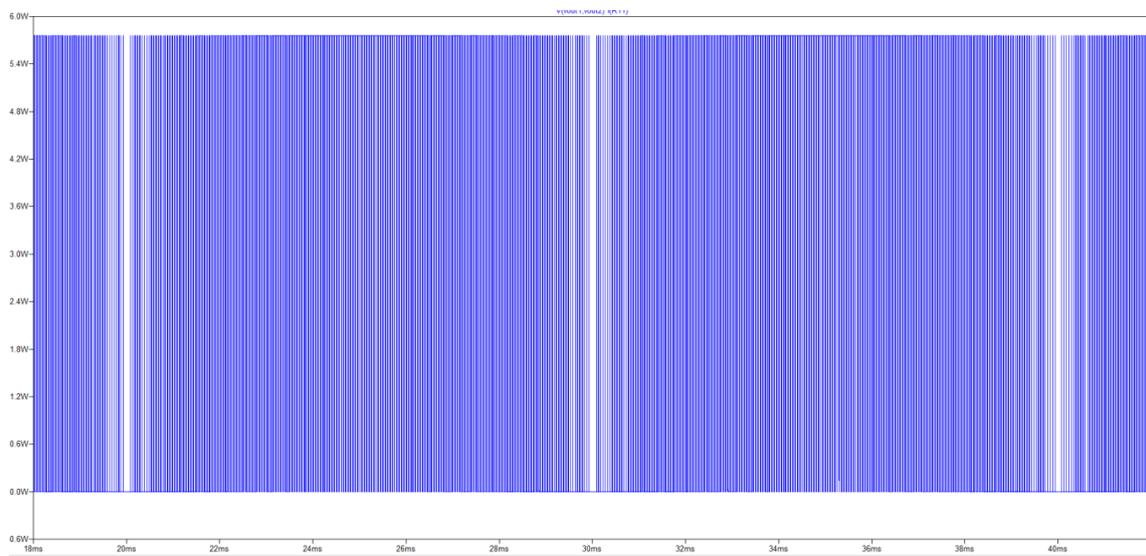


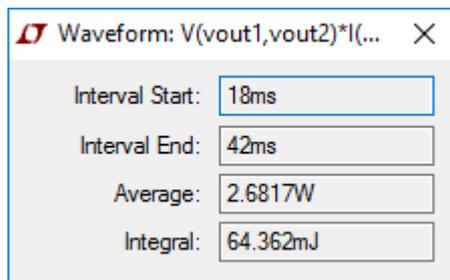
Figura 44: Potencia de la carga resistiva, Puente en H.



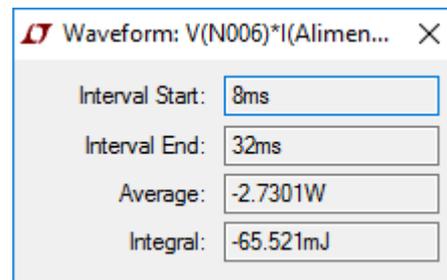
Figura 45: Potencia de los mosfet, Puente en H, carga resistiva.



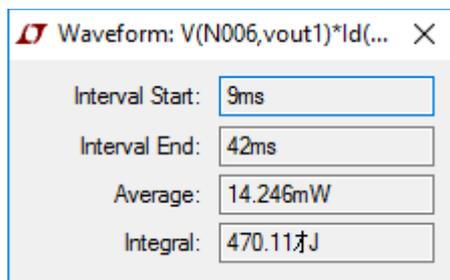
Figura 46: Potencia de la fuente de alimentación, Puente en H carga resistiva.



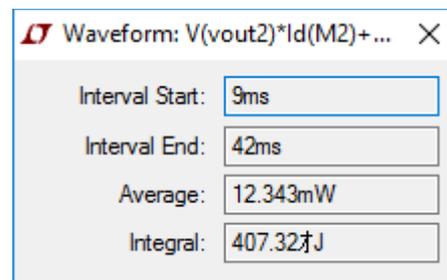
(a)



(b)



(c)



(d)

Figura 47:

- (a) Valores promedio y RMS de la salida
- (b) Alimentación
- (c) Mosfets superiores
- (d) Mosfets inferiores

Posterior a esta simulación se hizo una equivalente pero con una carga inductiva en paralelo a un condensador de $300\mu\text{F}$.

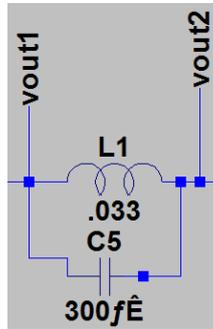
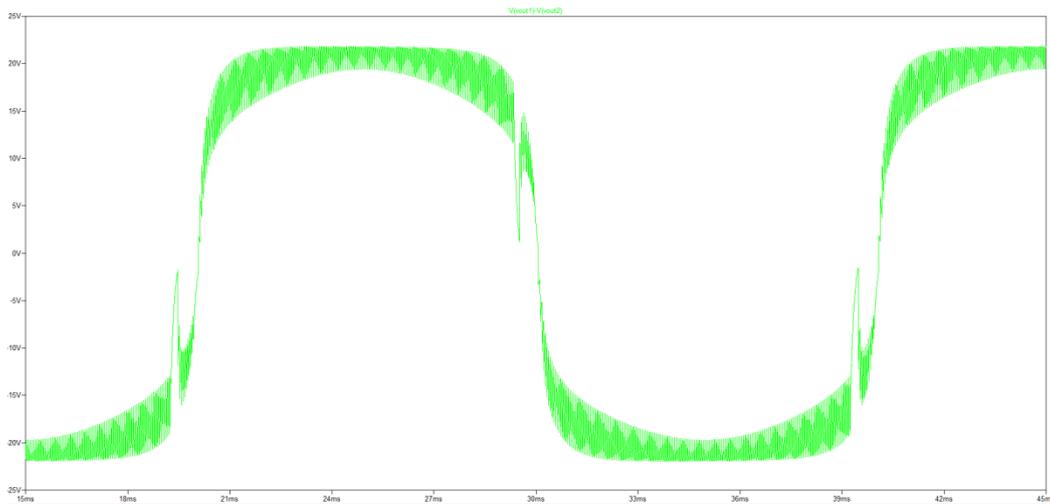
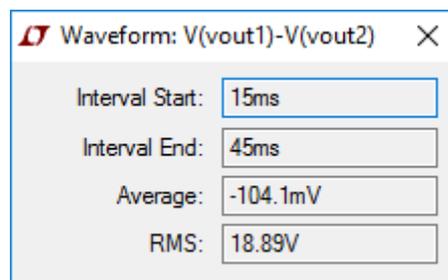


Figura 48: Esquema de carga inductiva y filtro por condensador.

La tensión de salida toma una forma muy parecida a una señal senoidal pura al colocar el condensador, cuanto más grande sea el condensador más se parecerá a una señal cuadrada. Como se observa en la figura 49 sigue existiendo una oscilación en la parte superior e inferior del ciclo en la que si se intentara tomar el valor medio de esta zona, se tendría una senoidal pura.



(a)

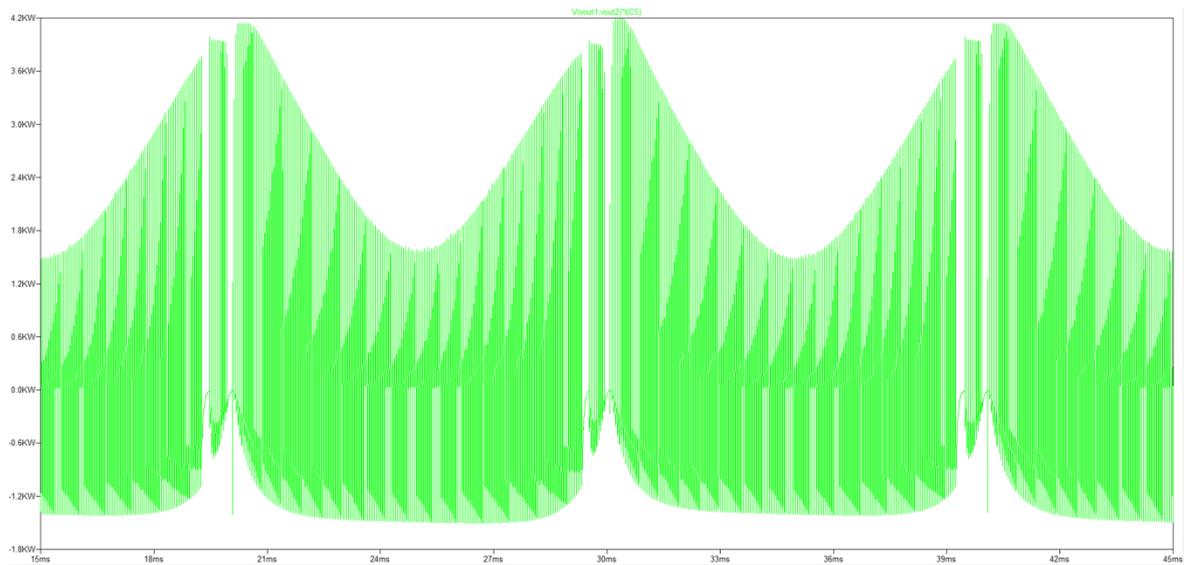


(b)

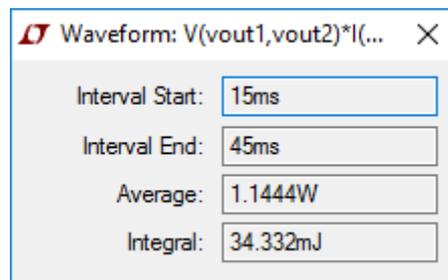
Figura 49:

- (a) Gráfica de la tensión de salida.
- (b) Tabla con los valores promedio y eficaz con carga inductiva.

El condensador colocado en paralelo se convierte en el segundo componente que consume una cantidad de potencia considerable que en este caso sigue siendo despreciable con respecto a la potencia cedida a la carga.



(a)

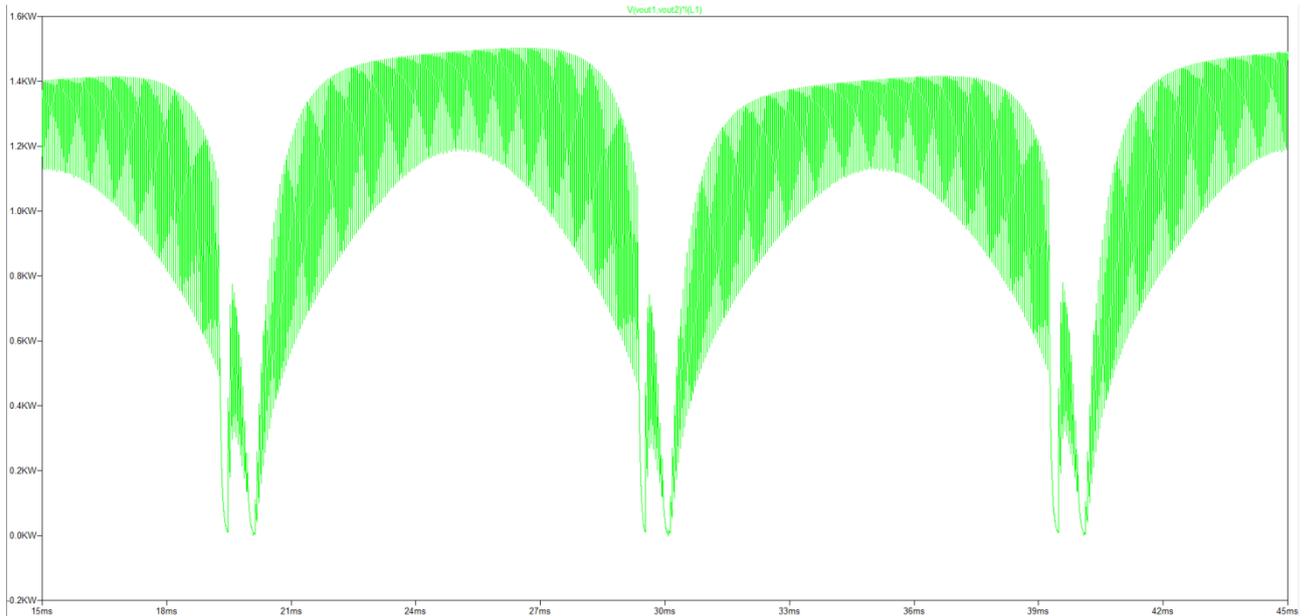


(b)

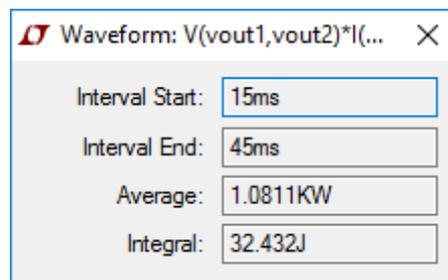
Figura 50:

- (a) Gráfica de la potencia del condensador
- (b) Tabla con los valores promedio e integral con carga inductiva.

Si se compara la potencia de salida de la figura 51 con la potencia de la alimentación de la figura 52 se tiene que más del 90% de la potencia media de la alimentación se transmite a la carga y si se atiende en la integral, se refleja que este valor disminuye hasta el 75% de la potencia.



(a)



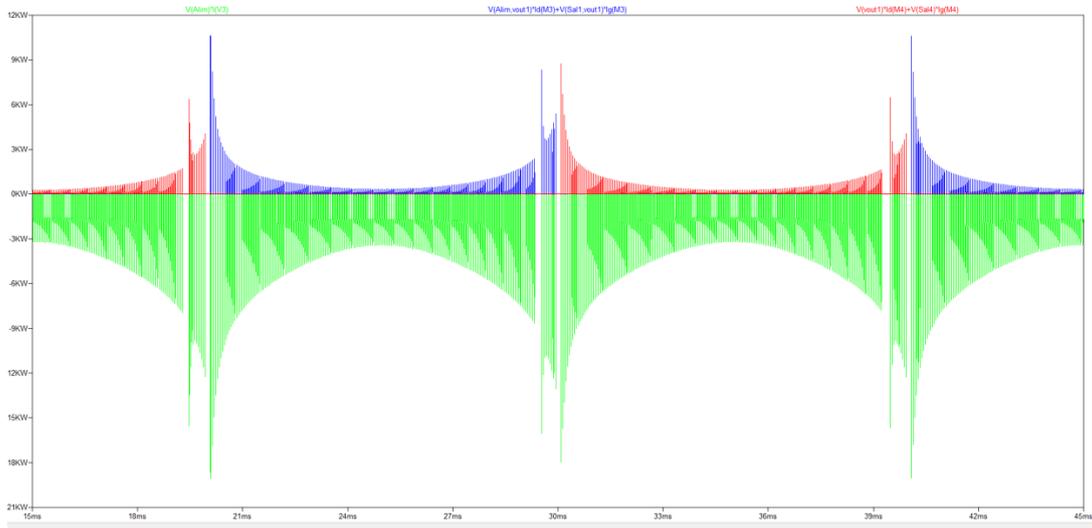
(b)

Figura 51:

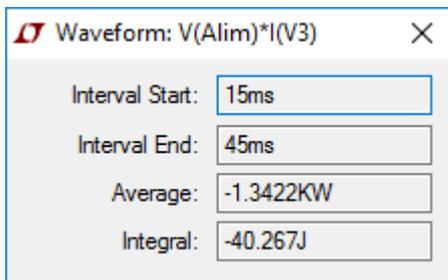
(a) Gráfica de la potencia de salida

(b) Tabla con los valores promedio e integral con carga inductiva.

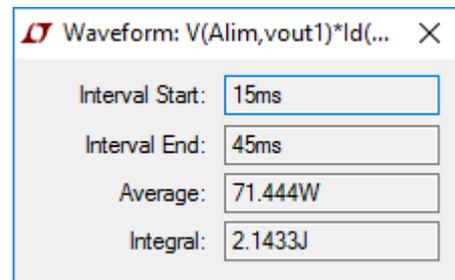
Con la carga inductiva se nota un aumento muy sustancial en la potencia consumida por los mosfet que toma casi el 10% de la potencia del sistema. A medida que se aumenta la tensión de entrada también aumenta el consumo de potencia en los mosfets y el condensador.



(a)



(b)



(c)

Figura 52:

- (a) Gráfica de la fuente de alimentación (verde) y la potencia disipada en los mosfet (azul y rojo)
- (b) Tabla con los valores promedio e integral con carga inductiva de la alimentación
- (c) Tabla con los valores promedio e integral con carga inductiva de los mosfet

En la figura 53 se muestra una fotografía del montaje del Puente en H, donde se ha desplazado la salida a través de cables a un punto inferior para poder conectar varios tipos de carga sin tener que interferir con el cableado del puente. Para el montaje se tuvo que soldar trozos más finos a las patas de los diodos, ya que estas eran más gruesas que los orificios de la protoboard y sustituir los cables que fueran de varios filamentos por otros de un solo núcleo.

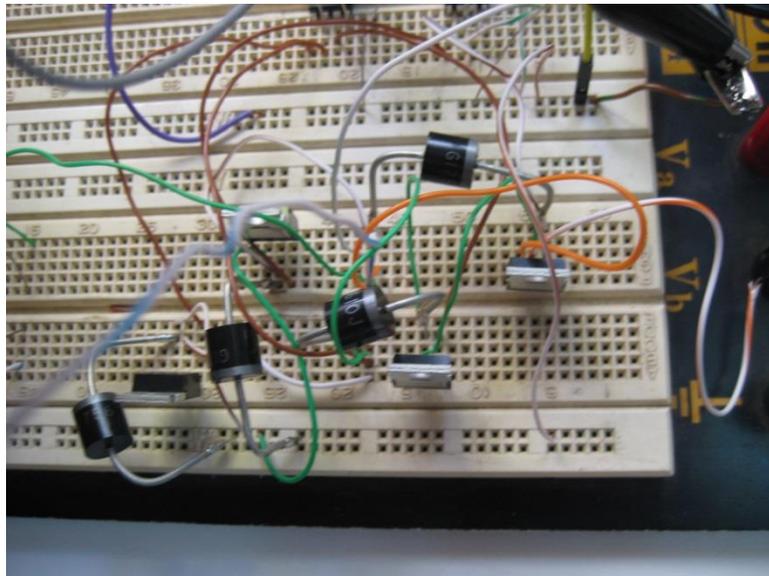


Figura 53: Fotografía del montaje del Puente en H sin carga.

La señal de salida resulta ser la esperada y es similar a la que se obtuvo en la simulación con la diferencia de que los pulsos de la señal lógica tienen los pulsos más anchos en la parte superior en la simulación y este hecho se refleja en la salida.



Figura 54: Fotografía de la tensión de salida con una carga resistiva.

Se colocó un condensador de $2.2\mu\text{F}$ obteniendo la señal en la salida de la figura 55 que muestra una forma casi cuadrada donde si se hace el promedio de la zona oscilante se obtiene una forma senoidal pura.

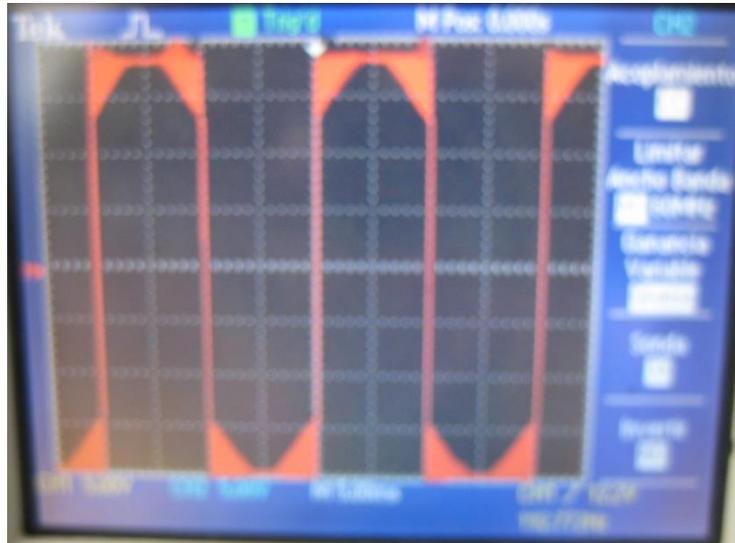


Figura 55: Fotografía de la tensión en la carga con un filtro por condensador.

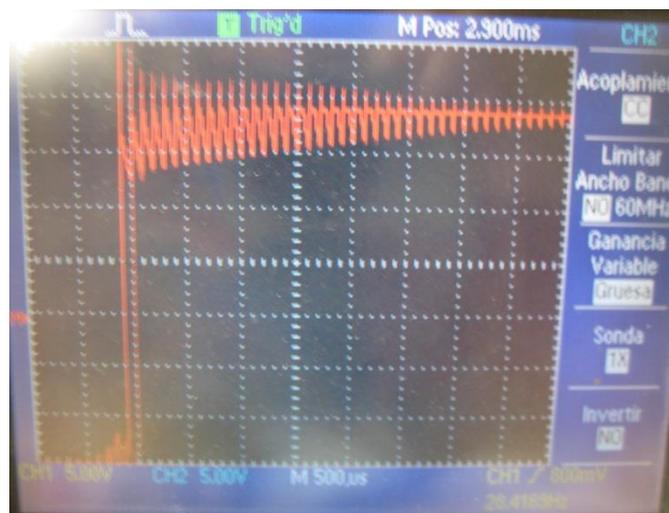


Figura 56: Fotografía ampliada de la oscilación de la señal de salida con condensador.

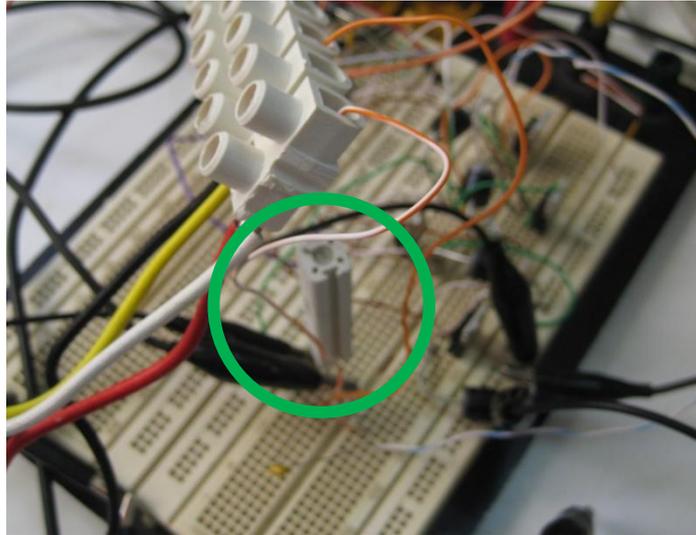


Figura 57: Fotografía que muestra la resistencia usada.

El partidor de tensión para la realimentación con Arduino Uno se diseñó con una resistencia de $10\text{M}\Omega$ colocada en la pata negativa de la carga conectada al pin analógico A0 y con otra resistencia de $100\text{k}\Omega$ colocada en la misma parte negativa colocada a tierra formando una división de la tensión entre 100 para que la salida pueda ser leída correctamente (entre 0 y 5 voltios). Se instalan resistencias elevadas para que no interfieran con la carga. Este sistema no llegó a implementarse debido a que solo se realizó una prueba a la tensión máxima y fue solo con la carga conectada. Si se comprobó que la realimentación funcionaba utilizando un osciloscopio para generar una señal alterna y se probó que operaba correctamente según lo programado.

Para la comprobación de funcionamiento del Puente en H, se utilizaron mosfets IRF630N que no pueden alcanzar los 310 voltios, pero de los que se tenía disponibilidad en el laboratorio para sustituir en caso de fallo.

4.4- Experimentos realizados.

El primer experimento consistió en colocar un transformador a la salida del puente para para comprobar que la señal de salida del puente puede ser usada en cargas inductivas y tenga una forma de onda adecuada.



Figura 58: Fotografía de transformador utilizado en la carga, plano alzado.



Figura 59: Fotografía de las especificaciones del transformador.

Se le colocó un ventilador inductivo de una potencia no muy alta para comprobar el sistema y un transformador para convertir la salida del puente de 24V alimentada con fuentes de alimentación a 220V RMS que precisa el ventilador (figura 61). Se midió tanto con multímetro como con un osciloscopio la forma de onda de la salida resultando que efectivamente se tenían alrededor de 220V RMS y el ventilador funcionaba, pero la señal de onda se distorsionaba formando zonas oscilantes en los puntos de mayor y menor tensión. A este circuito se le añadió una batería de condensadores a modo de filtro de 0.1mF de capacidad, que es un condensador de gran capacidad. El resultado fue una mejoría considerable de la forma de onda, donde se observaba prácticamente senoidal (con algún pequeño pico), pero con el inconveniente de que se añadía mucho consumo a la fuente de alimentación pasando de extraer 1.5A a casi 2.3A al conectar la batería de condensadores.



Figura 60: Fotografía del ventilador utilizado como carga inductiva.



Figura 61: Fotografía de las especificaciones del ventilador.

El siguiente experimento consistió en comprobar el funcionamiento del sistema a 110V RMS sin necesidad de un transformador en la salida. Para ello, se retiró el transformador en la carga y se colocó el circuito DC-DC por Fly-Back de Néstor para alimentar el Puente en H. Como carga se utilizó un ventilador de 110V RMS. Se hicieron cambios en el circuito elevador de continua en el que se cambió la relación de resistencias del partidor de tensión que servía de referencia para el TL404 para que mantuviera la señal alrededor de los 150V. También se sustituyeron los tres transformadores iniciales por dos transformadores en serie de 12V.

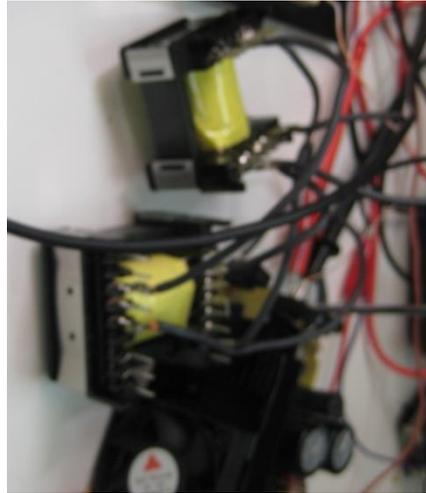


Figura 62: Fotografía de dos transformadores utilizados en serie en el laboratorio.

El resultado obtenido fue la generación de los 150V de continua a la salida del circuito DC-DC y el funcionamiento del ventilador. Se observó que si se pulsaba el botón de Reset del Arduino a la vez que se encendía la alimentación, el ventilador tardaba en ponerse en funcionamiento y que cuando empezaba a arrancar no empezaba a girar casi instantáneamente, sino que aumentaba progresivamente en un muy corto periodo de tiempo.

El experimento se dejó en funcionamiento durante más de 10 minutos sin que ningún elemento sufriera un aumento de temperatura significativo, salvo los mosfet del elevador DC-DC. Sin embargo, el disipador evitaba que estos llegasen a temperaturas dañinas, y si fuera necesario, se podría conectar el ventilador del mismo.



Figura 63: Montaje de ventilador de 110 V RMS utilizado en el laboratorio con regleta para encender la carga

También se realizó un experimento para comprobar si podían existir problemas en los transformadores por sobreesfuerzo.

Para comprobar si existía este efecto se colocó una resistencia muy pequeña (1Ω) en serie con el polo negativo de la alimentación del circuito DC-DC. Al medir en esta resistencia se obtenía la intensidad de la alimentación. Con el osciloscopio se modificó el resultado para visualizar el promedio, pero se seguía sin obtener un dato que fuera útil por lo que se procedió a integrar dicha señal. Para ello, se utilizó el circuito inversor de la figura 64.

El circuito consiste en separar la corriente con un AO en configuración de seguidor de tensión pasarlo por un diodo y colocar un paralelo de una resistencia y un condensador y jugar con los valores de estos últimos para obtener un efecto de integración.

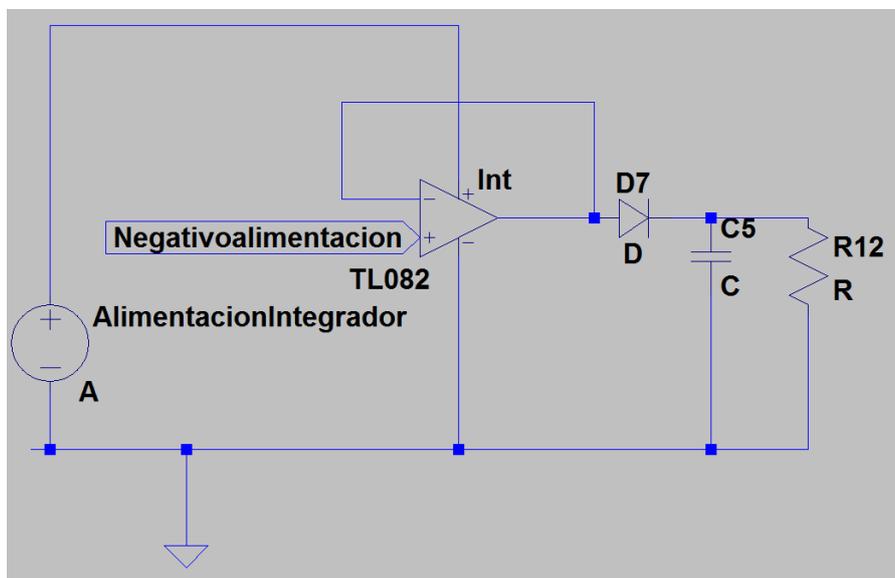


Figura 64: Esquema del integrador.

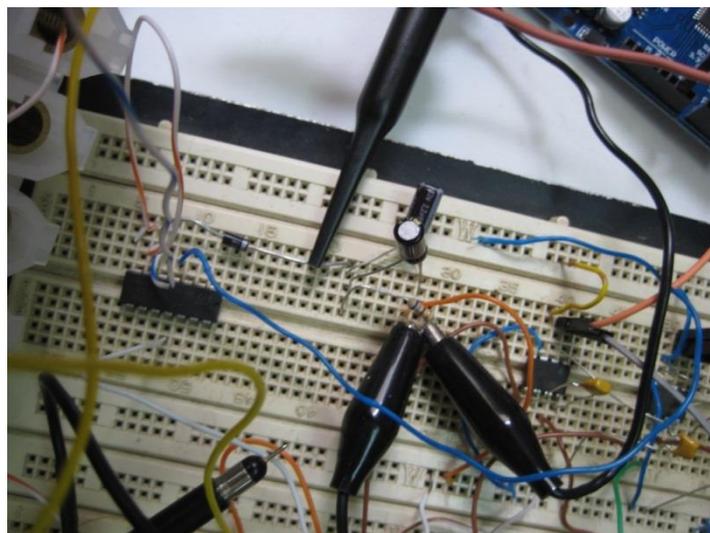


Figura 65: Fotografía del montaje de circuito integrador.

La señal observada al medir en la salida del integrador con el osciloscopio muestra una señal con mucho ruido pero de valores aceptables, ya que la señal no presentaba ruido. Lo que se observa es el efecto que provocan los transformadores en la corriente generando medio amperio de consumo adicional.

En la figura 66 se puede observar que hay un periodo de picos mayores que en el resto de la onda, este efecto es debido al arranque del sistema y la carga de condensadores del circuito DC-DC.

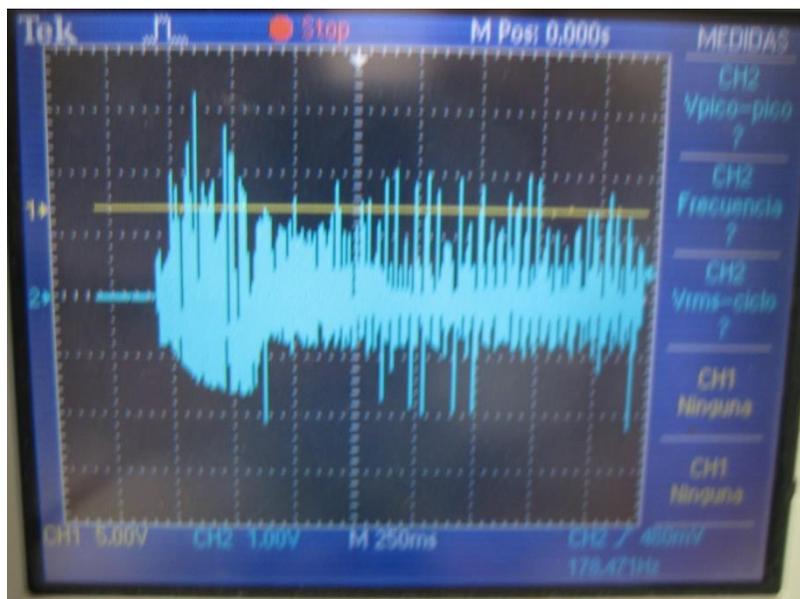


Figura 66: Fotografía que muestra la señal en el integrador y el pico de arranque.

Reduciendo el valor de la alimentación el consumo se mantenía constante hasta que se alcanzaban los 10V, a partir de este valor también disminuía la corriente de la fuente de alimentación por lo que se reducía el consumo. Cuando se restringía la tensión de salida por debajo de los 7V la tensión de salida del elevador DC-DC empezaba a disminuir.

Al medir otra vez en el integrador se generó un cambio drástico donde se redujo la amplitud de la señal, pero como se ve en la figura 67 esto desencadenó un ruido armónico de una frecuencia muy baja (el osciloscopio se encuentra a 2.5s por división).



Figura 67: Fotografía que muestra la salida del integrador en más de 10 segundos.

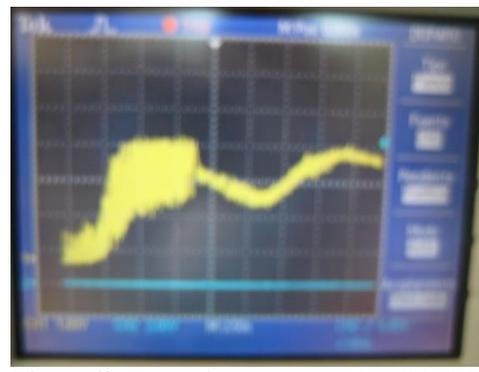


Figura 68: Fotografía que muestra la salida del integrador mientras se resetea el Arduino.

Por último, se sustituyó el ventilador de 110V RMS por el de 230V RMS, y se probó con los únicos mosfets que se tenían en el laboratorio que soportasen más de 310 V de pico; cuatro mosfet P9NB60. Al arrancar el sistema, no se generaba tensión en el elevador DC-DC, indicio de un cortocircuito en el Puente en H y al comprobar cada uno de los mosfets se descubrió que estos se habían dañado. Se desconoce si por el circuito en uso o por defectos derivados de haber sido usados anteriormente. Debido a que no hay más existencias de estos mosfets o equivalentes en el laboratorio y no se han conseguido otros a tiempo para la resolución de este experimento, el experimento finaliza por falta de material.

5- PRESUPUESTO

5.1- Presupuesto de los circuitos.

Para el desarrollo de este presupuesto se añadirá el presupuesto del circuito DC-DC desarrollado por por Néstor y se realizará la suma de ambos circuitos al final de este apartado

5.1.1 Circuito DC-DC

DESCRIPCIÓN	CANTIDAD	PRECIO UNITARIO(€)	PRECIO TOTAL(€)
Disipador de calor	1	3.85	3.85
Placa perforada para soldar	1	2.40	2.40
Cable eléctrico unipolar	2	0.54	1.08
Condensador 47 μ F 50V	3	1.77	5.31
Condensador 1000 μ F 63V	4	1.386	5.544
Condensador 470 μ F 200V	2	0.107	0.214
Condensador 100 μ F	1	0.14	0.14
Condensador 0.1 μ F	4	0.65	2.60
Condensador 1 μ F	2	0.70	1.40
Resistencia	20	0.14	2.80
Diodo 1N4007 a 1000V	5	0.34	1.70
Condensador	1	0.345	0.345
TL494	1	0.463	0.467
Driver MOSFET 4451YN	2	1.4	2.80
Transformador de fuente de alimentación conmutada	3	8.09	24.27
TOTAL:			54.96

5.1.2- Circuito Driver.

DESCRIPCIÓN	CANTIDAD	PRECIO UNITARIO(€)	PRECIO TOTAL(€)
UNO R3 ATmega328P USB CH340 Compatible to Arduino UNO	1	3.49	3.49
Controlador de puerta HV H-Bridge driver L6387E	2	1.46	2.92
Condensador 0.1 μ F 500V	2	1.15	2.30
Condensador 1 μ F 50V	2	0.405	0.81
TOTAL:			9.52

5.1.3- Circuito puente en H y componentes adicionales

DESCRIPCIÓN	CANTIDAD	PRECIO UNITARIO(€)	PRECIO TOTAL(€)
Diodo de Potencia GI756 600V 6A	4	0.497	1.99
MOSFET N-Ch 600 V 10 Amp STP10NK60Z (equivalente a P9NB60)	4	1.26	5.04
Placa protoboard 82x53mm	1	5.15	5.15
Resistencia 1kΩ	1	0.021	0.021
Resistencia 100kΩ	1	0.061	0.061
Cable electrico unipolar 2.5mm	2(metros)	0.28	0.56
TOTAL			12.82

Los demás elementos utilizados en el desarrollo tales como fuentes de alimentación y componentes usados para las pruebas no se contemplan ya que no forman parte del circuito en conjunto.

5.2- Mano de obra

DESCRIPCIÓN	HORAS DE TRABAJO	PRECIO POR UNIDAD(€)	PRECIO TOTAL(€)
Desarrollo y diseño	300	8	2400

5.3- Coste total.

DESCRIPCIÓN	PRECIO(€)
Circuito DC-DC	54.96
Circuito Driver	9.52
Circuito Puente en H y adicionales	12.82
Total coste circuito completo	77.30
Mano de obra	2400
TOTAL	2477.30

Se estima que el proyecto tiene un coste total de **dos mil cuatrocientos setenta y siete euros con treinta céntimos**.

6- CONCLUSIÓN

6.1- Comparativa comercial

A la hora de comparar el producto con los que ya existen en el mercado hay que tener en cuenta una serie de características (potencia, eficiencia, precio, disponibilidad, tensión de entrada y tensión de salida, entre otras). La característica más importante de este proyecto es que los inversores de 36V son poco comunes, los dispositivos que se venden suelen ser de 12, 24, 42 y 64 voltios. Suelen saltar de los 24 a los 42 debido a que no es muy habitual tener fuentes de 36V de alimentación, de hecho en nuestro circuito podemos reducir la tensión no necesitando que sea exactamente de 36V, por lo que tenemos un margen de la tensión de entrada.

El segundo aspecto en el que destaca este sistema es el arranque suave. En el mercado se pueden encontrar algunos productos que vienen con la especificación de tener un sistema Soft-Start, pero no se parece a nuestro sistema de arranque lineal. Lo que hacen realmente es controlar la corriente de salida para que no se conecte instantáneamente la carga, en este proyecto se empieza con una tensión y frecuencia nula y se aumenta paulatinamente.

La mayor ventaja que tiene el circuito es el precio, los inversores son bastante caros, variando de precio desde cientos de euros a miles de euros. Es verdad que muchos de estos sistemas tienen displays y sistema de protección, pero estos sistemas podrían ser implementados y seguiría siendo más rentable diseñar el inversor que comprarlo. Esto se debe a que se ahorra en algunos costes fijos como: comprobaciones, rentas, transporte, etc., los cuales encarecen el precio drásticamente.

Hay que tener en cuenta, que no se ha comprobado satisfactoriamente a la tensión de diseño, pero sí para motores de 110V RMS, por lo que se podría utilizar sin problemas en motores de este tipo. El sistema no es fácil de trasladar, ya que está conectado por partes con cables y no es una base estable.

En definitiva, es un sistema funcional, que sirve como modelo para futuras mejoras.

6.2- Opinión personal

Este trabajo es el punto y final de un proceso de aprendizaje que es la carrera, ha requerido mucho tiempo para llegar hasta este punto, en el que considero que he podido demostrar los conocimientos y aptitudes adquiridos en el grado. Con la elaboración de este trabajo, llevo a la práctica competencias como la búsqueda de información y utilizarla correctamente para desarrollar este prototipo y hacer frente a las adversidades y darles respuesta.

Desde mi punto de vista; la parte más complicada y que más tiempo ha consumido de este trabajo ha sido la programación de Arduino y la simulación del circuito en LTSpice, que para alguien que esté acostumbrado al manejo y lenguaje puede ser trivial, para mí no lo fue. Al condensar todo el trabajo de un año en un informe, no se recoge material que se ha generado y debería también tenerse en cuenta, como el caso de vídeos, fotos

no tomadas, anécdotas y errores cometidos durante su desarrollo y es una pena que todo eso no pueda ser reflejado y así notarse la evolución a lo largo del avance de este trabajo final de grado.

7- REFERENCIAS Y BIBLIOGRAFÍA.

Artículo de “Diferencias entre los distintos tipos de inversores” (2017). Recuperado en MES de 2017, disponible en: https://www2.ineel.mx/proyectofotovoltaico/preg_20.html

Artículo de Fuente de Alimentación en Wikipedia (2017). Recuperado en MES de 2017, disponible en: <https://es.wikipedia.org/wiki/Mult%C3%ADmetro>.

Artículo de Multímetro en Wikipedia (2017). Recuperado en Abril de 2017, disponible en: <https://es.wikipedia.org/wiki/Mult%C3%ADmetro>.

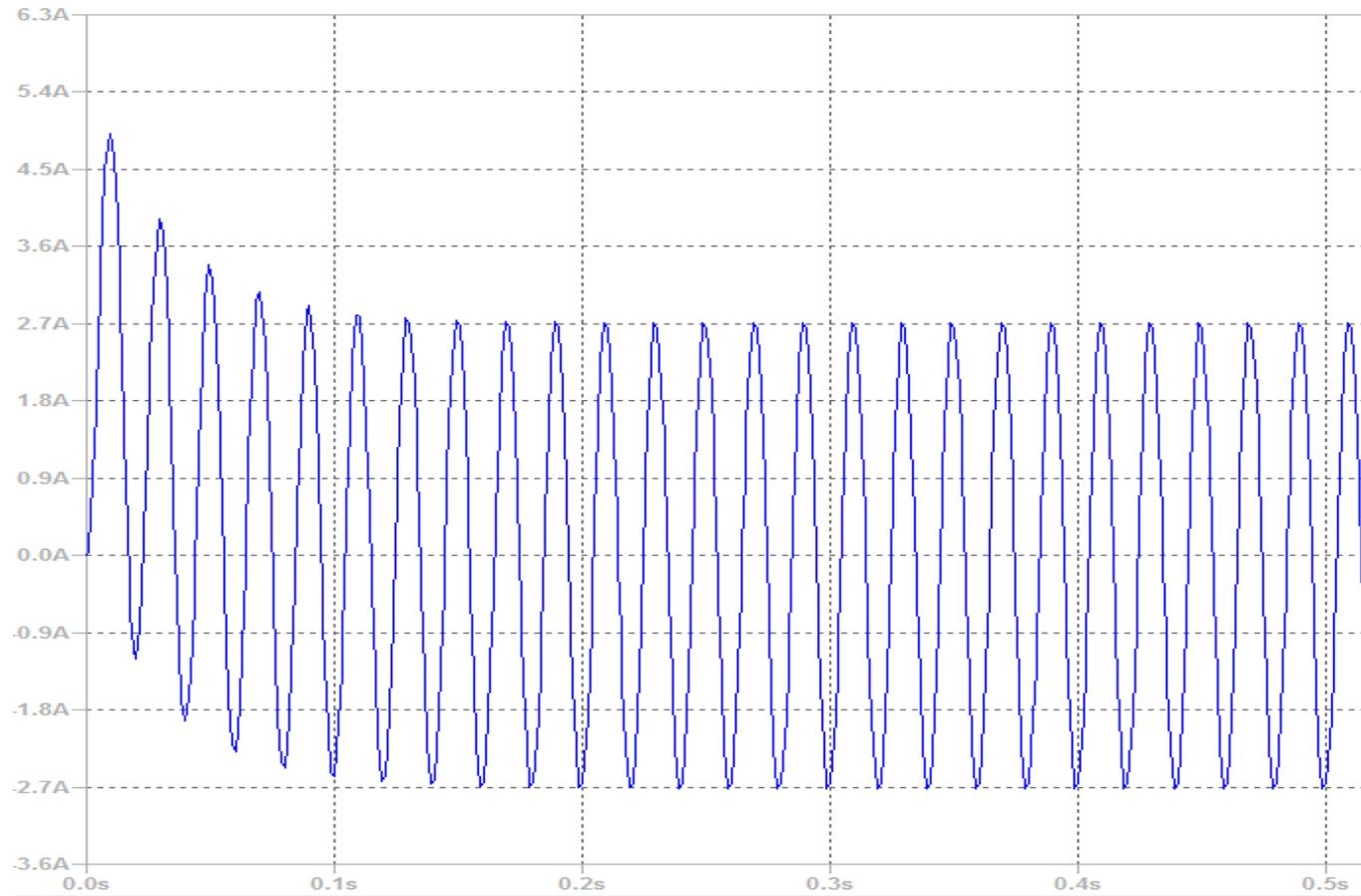
Artículo Osciloscopio en Wikipedia (2017). Recuperado en Abril de 2017, disponible en: <https://es.wikipedia.org/wiki/Osciloscopio>.

Artículo Placa de Pruebas en Wikipedia (2017). Recuperado en Abril de 2017, disponible en: https://es.wikipedia.org/wiki/Placa_de_pruebas.

Hart, D. W. (2001). *Electrónica de Potencia*. Madrid: Pearson Education, S.A.

ANEXOS

PLANOS, ESQUEMAS Y GRÁFICAS



Place a new Inductor on the schematic.

Figura 18: Gráfica de corriente de arranque en condiciones favorables.

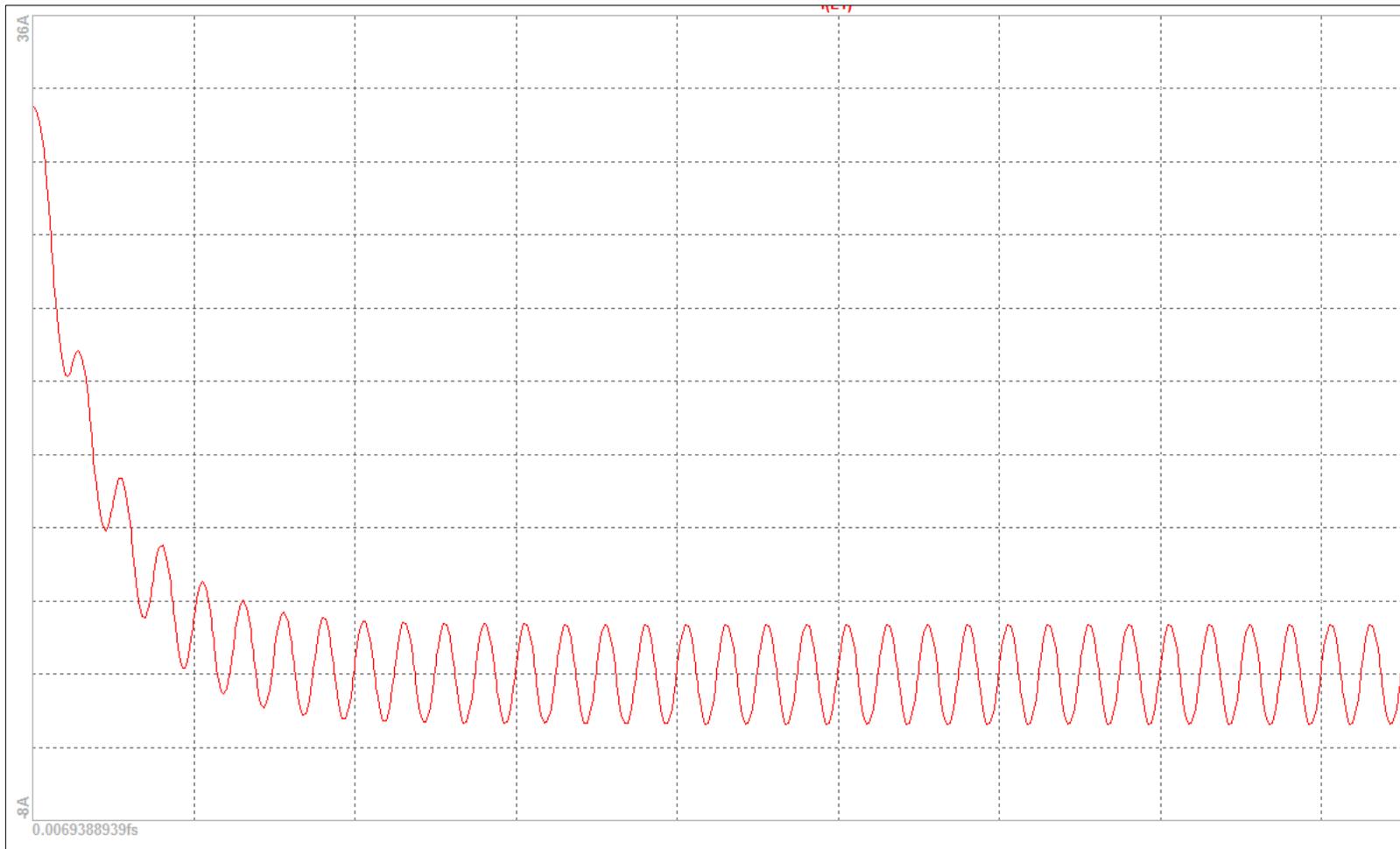


Figura 19: Gráfica de corriente de arranque en condiciones desfavorables.

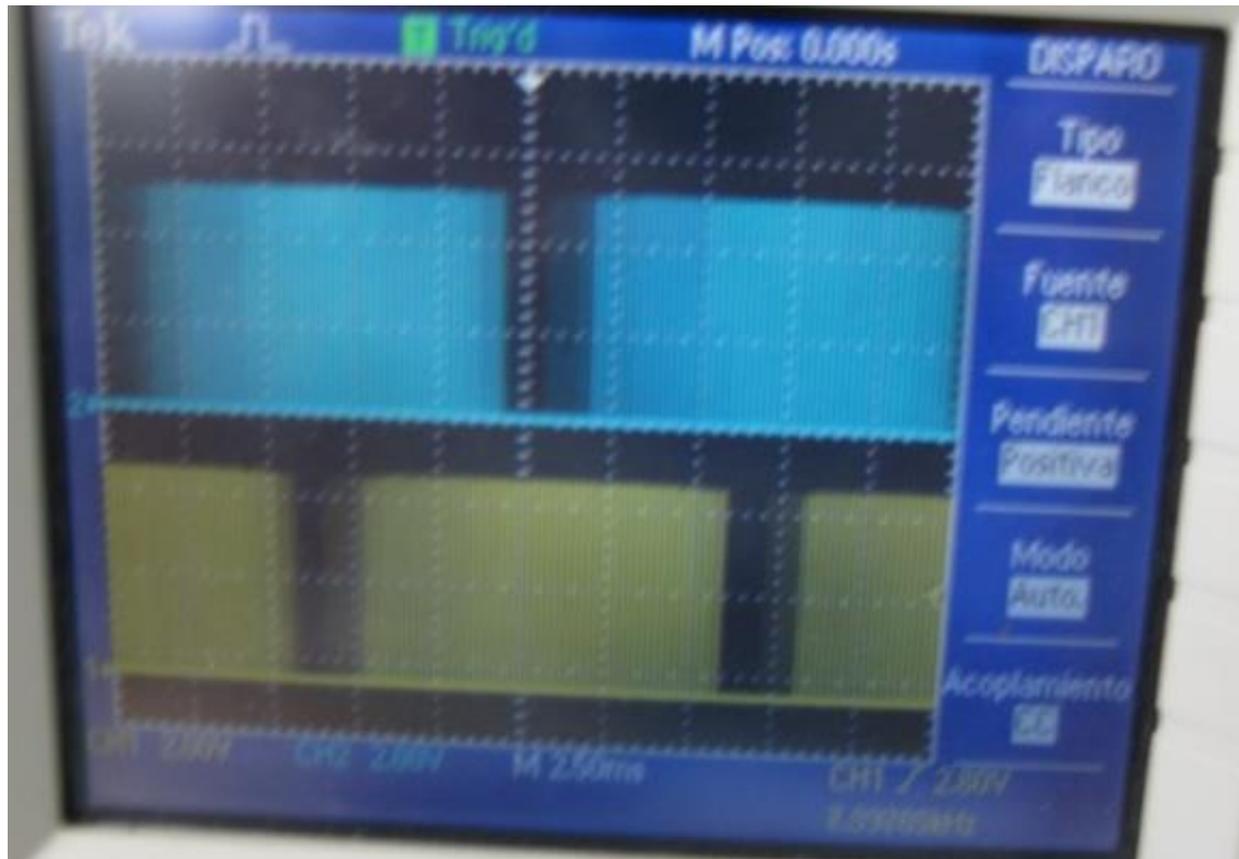


Figura 22: Señal SPWM de la salida del Arduino Uno, la señal tiembla.

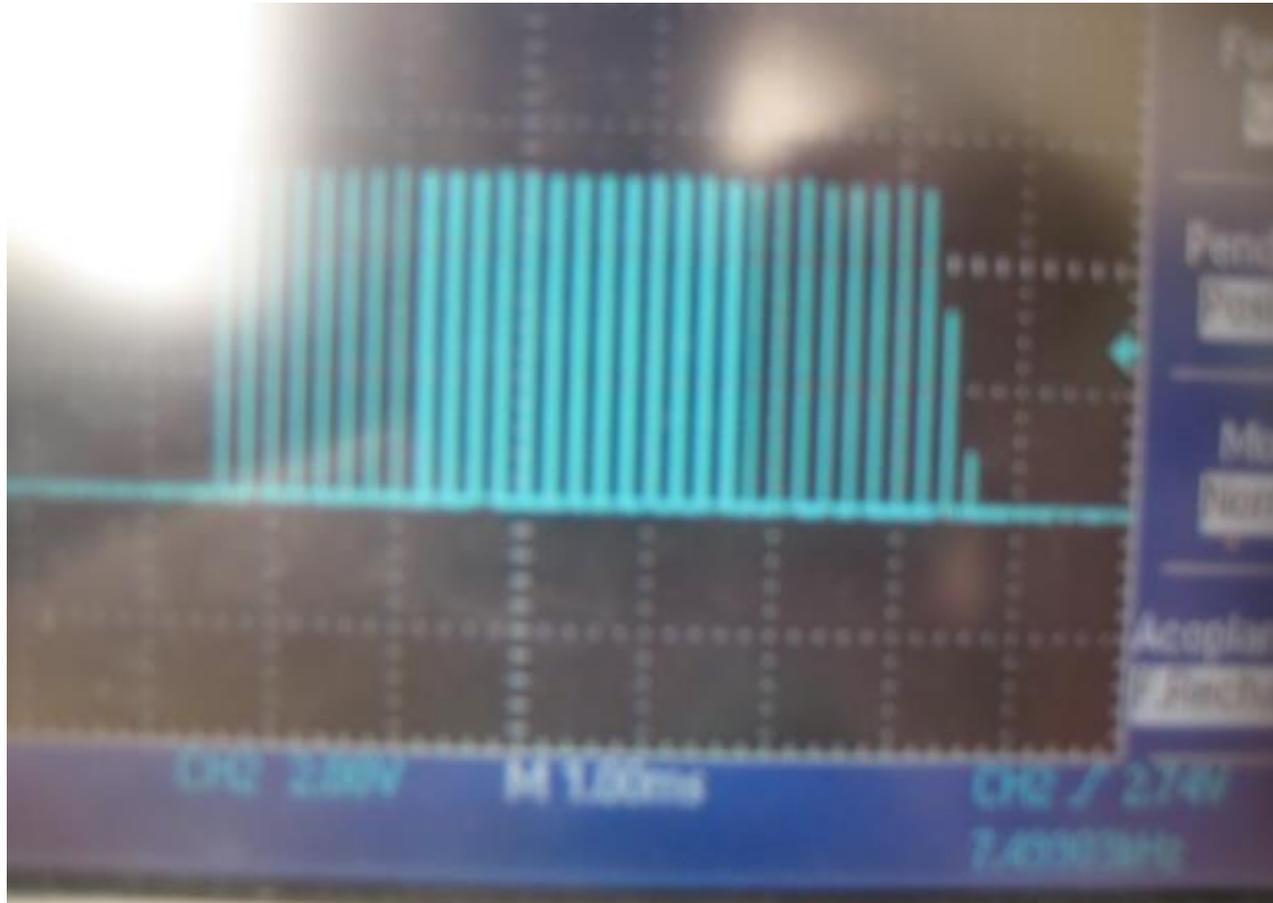


Figura 23: Ampliación de un pulso de la señal SPWM

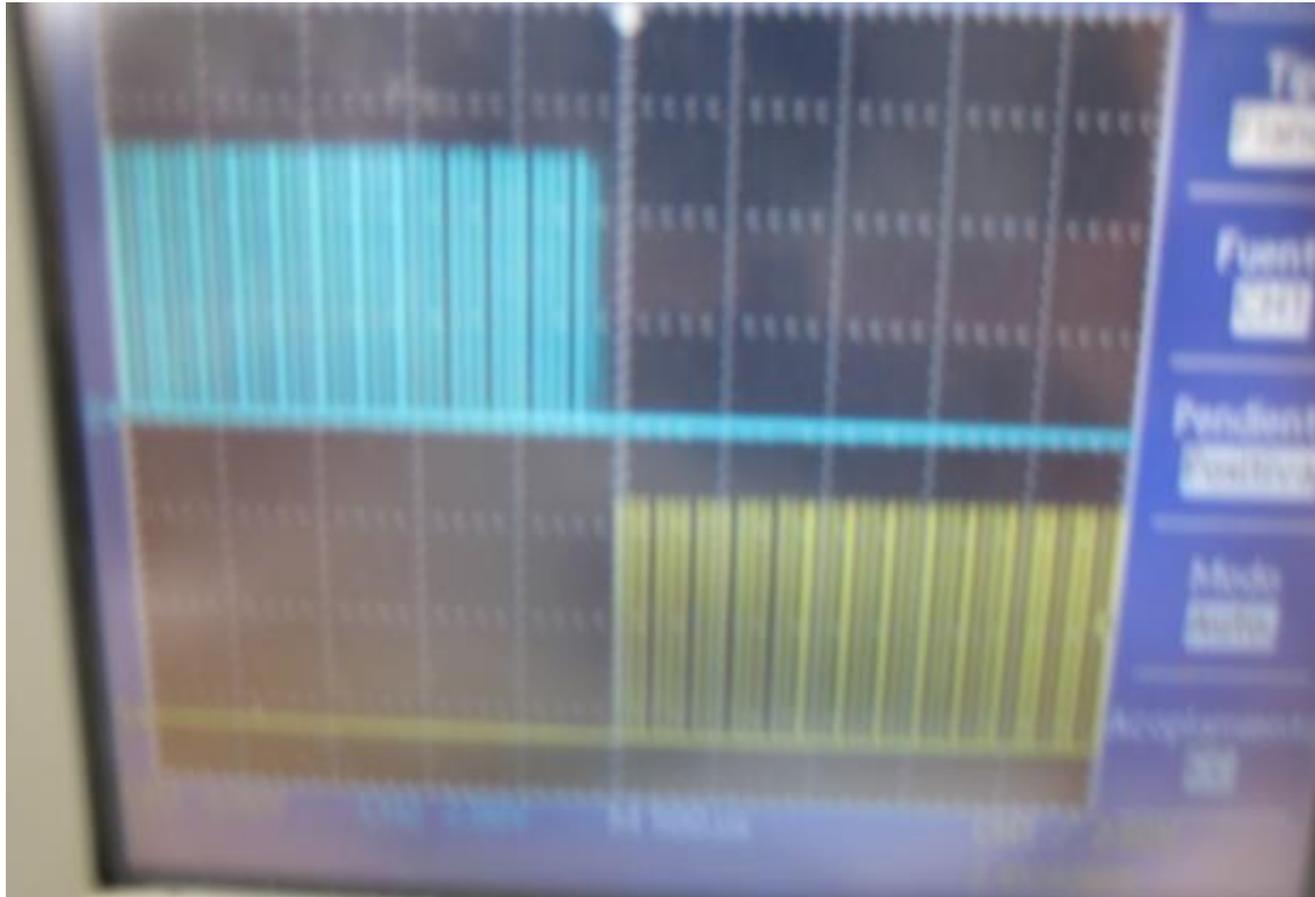


Figura 24: Ampliación de cómo encajan las dos señales SPWM, con trigger correcto.

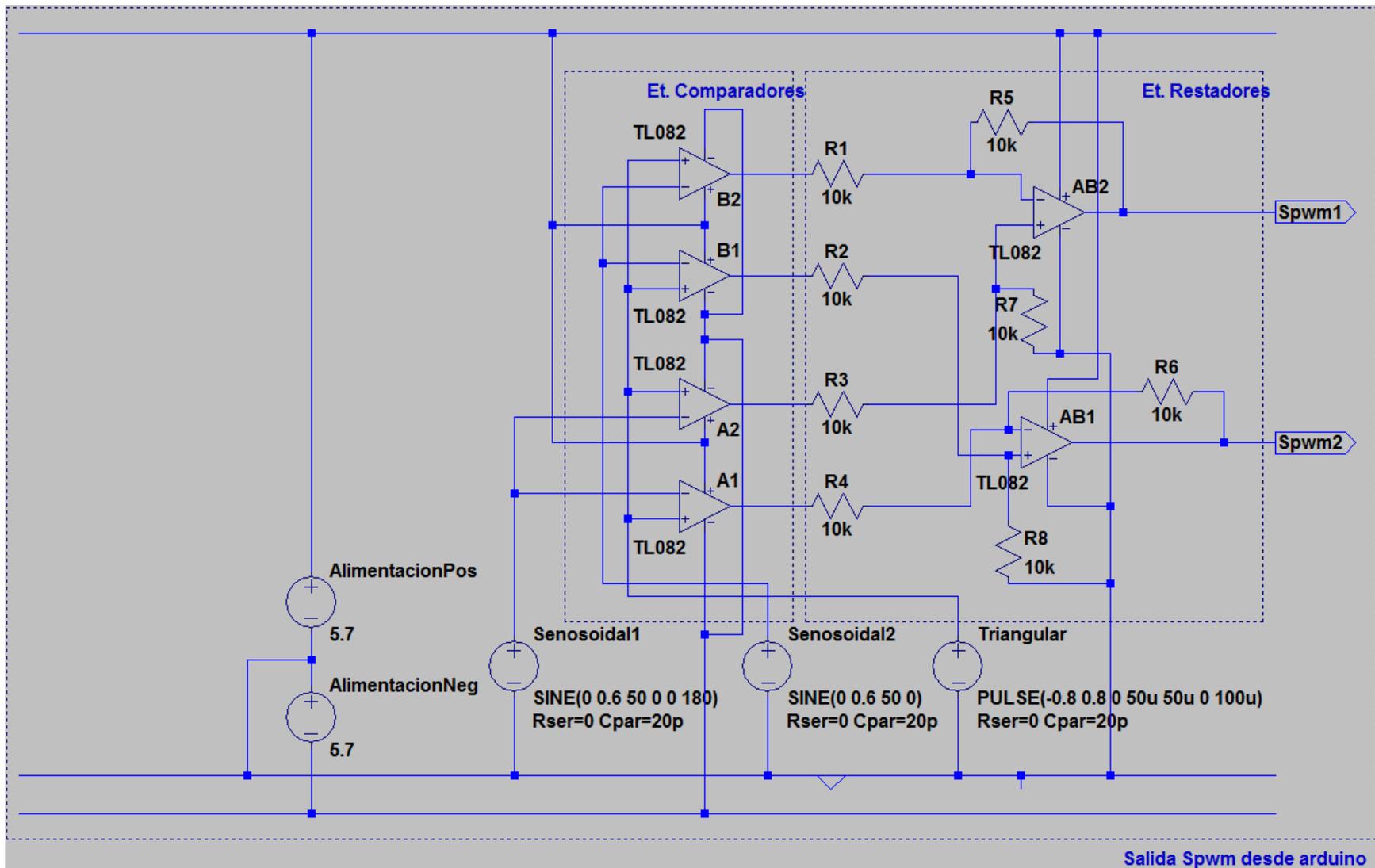


Figura 32: Esquema de simulación de salida Spwm en LTSpice

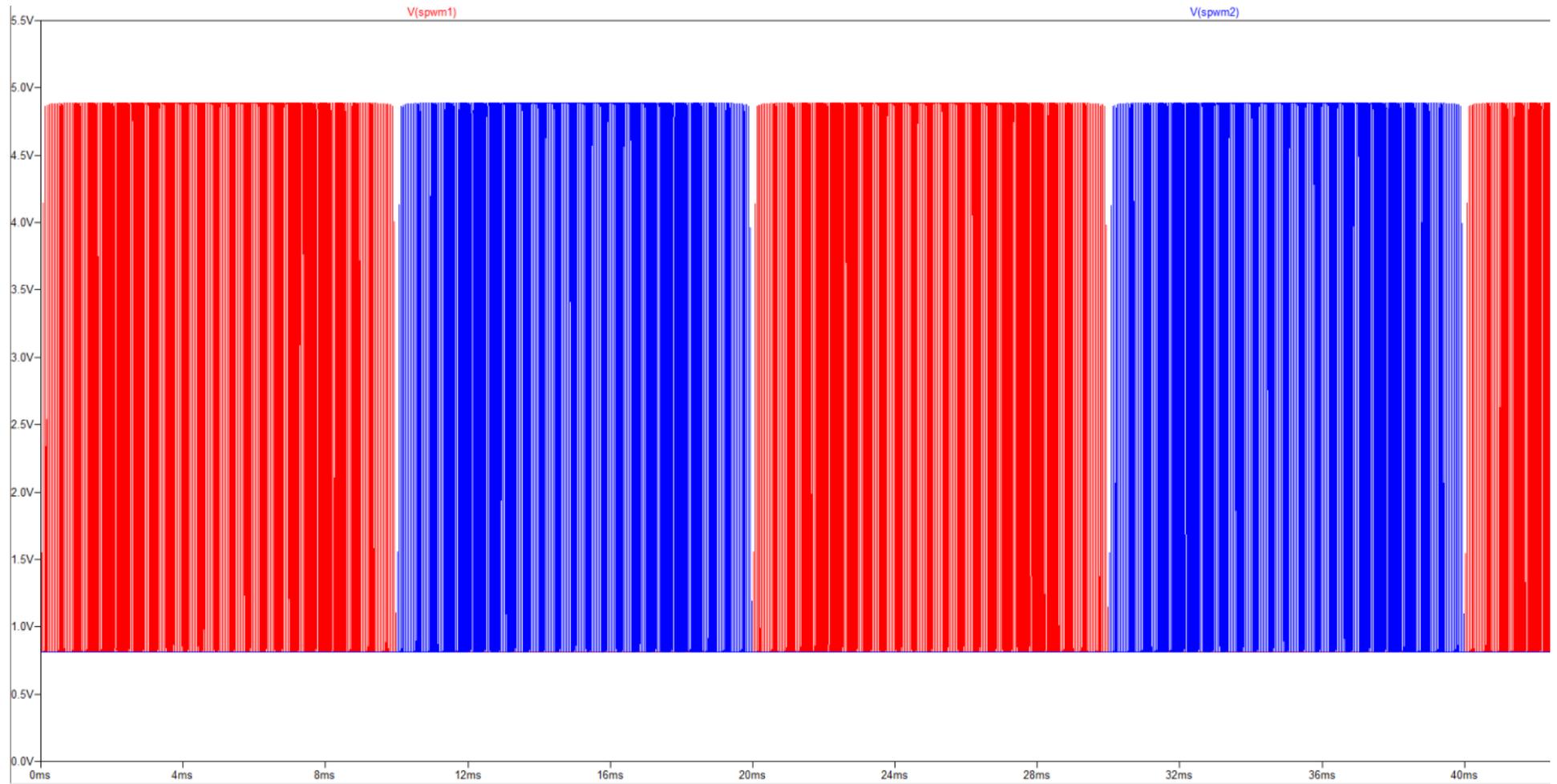


Figura 33: Simulación de la salida del Arduino en LTSpice.

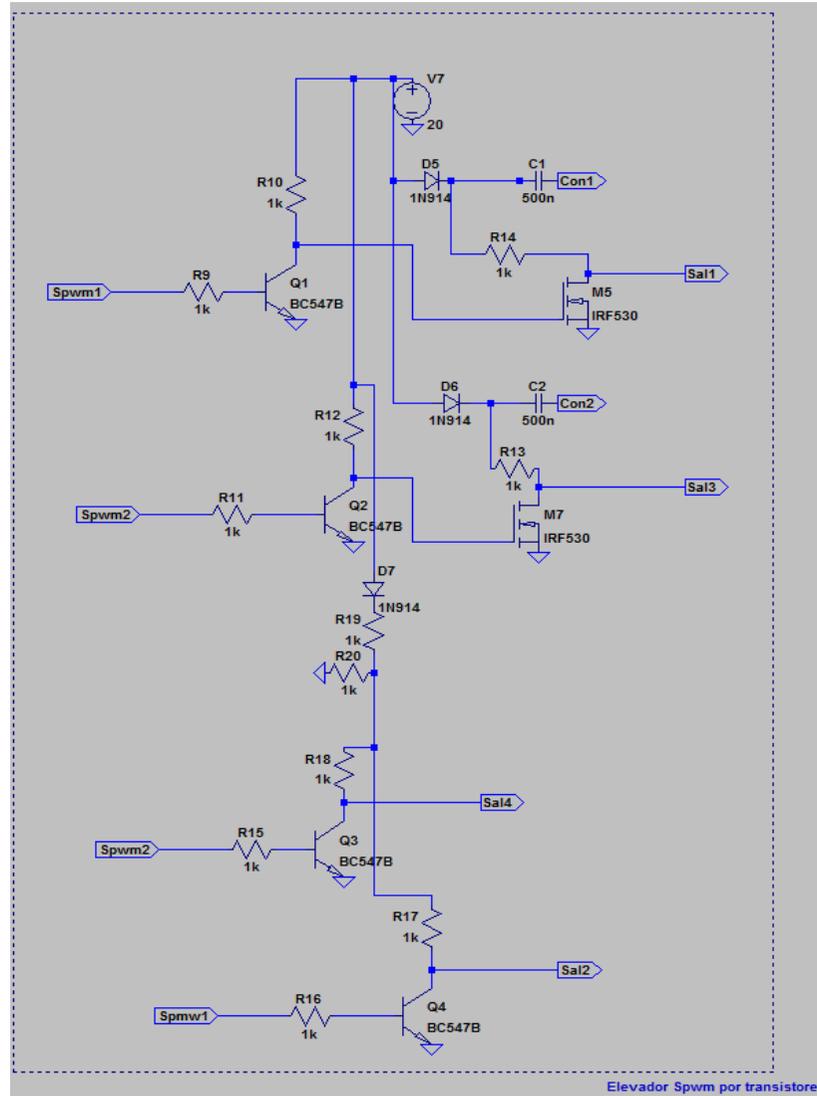


Figura 34: Esquema de elevador por transistores en LTSpice.

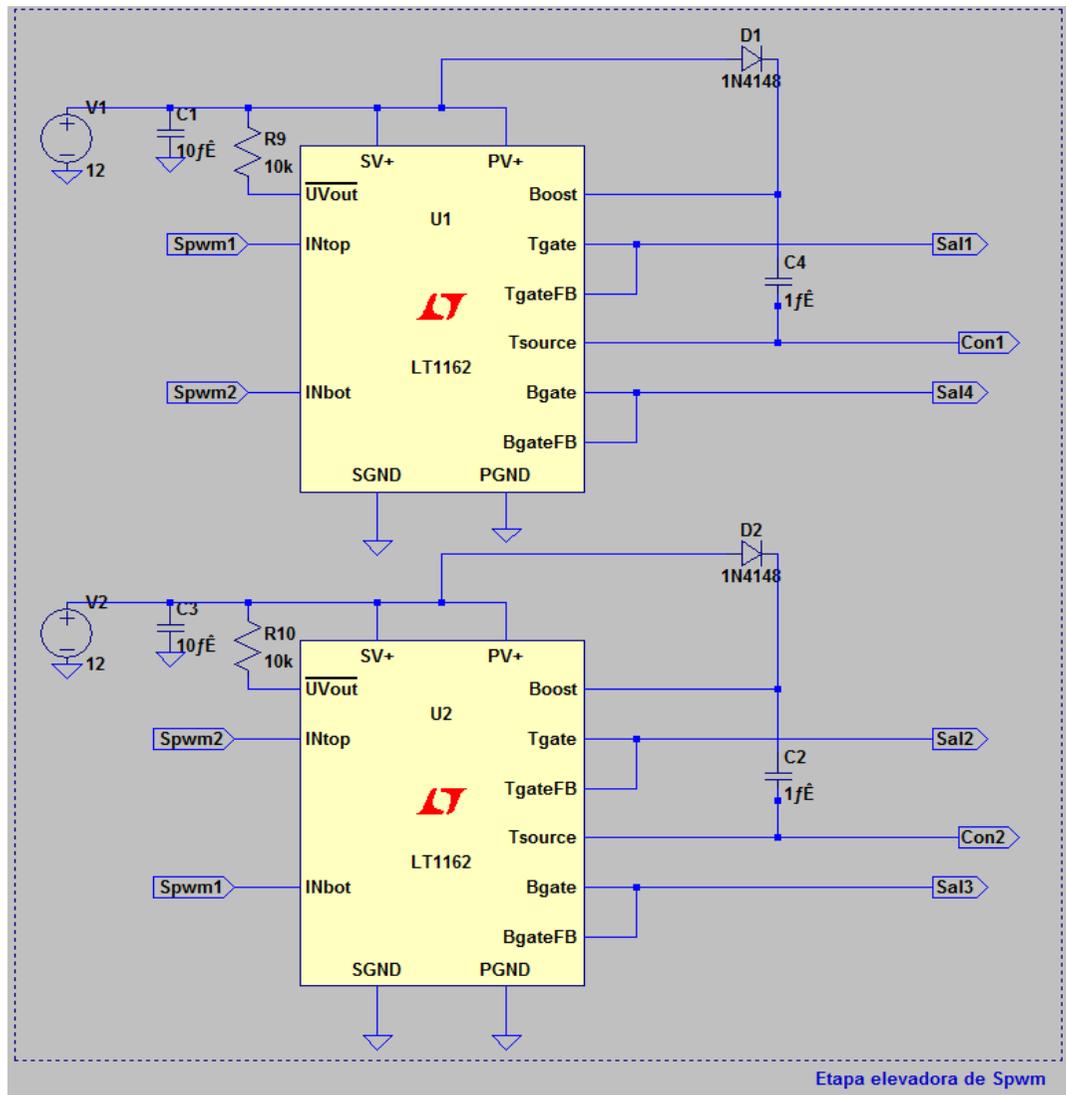


Figura 35: Esquema etapa elevadora de Spwm en LTSpice.

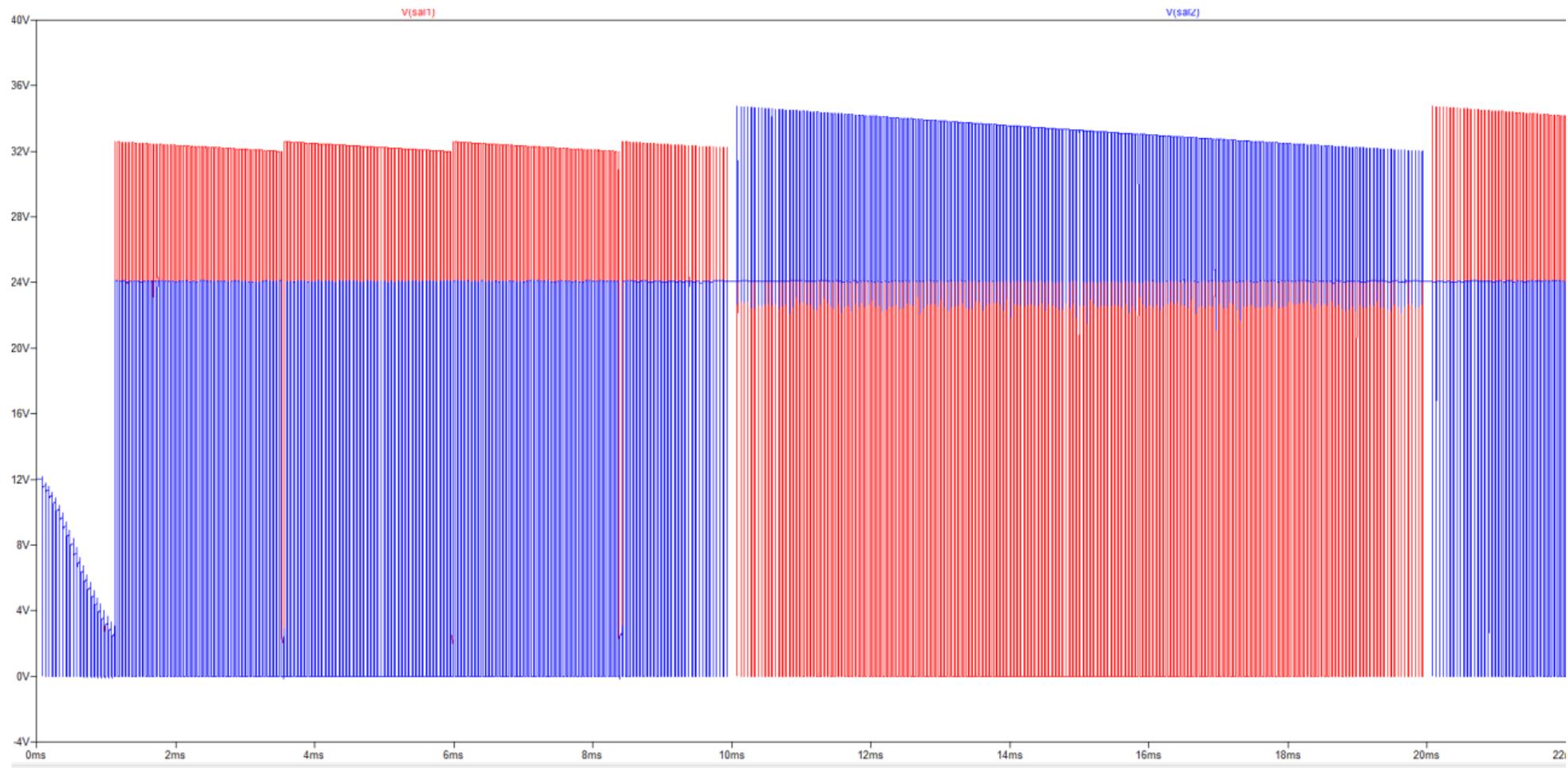


Figura 36: Voltaje de entrada a la puerta de los mosfets superiores.

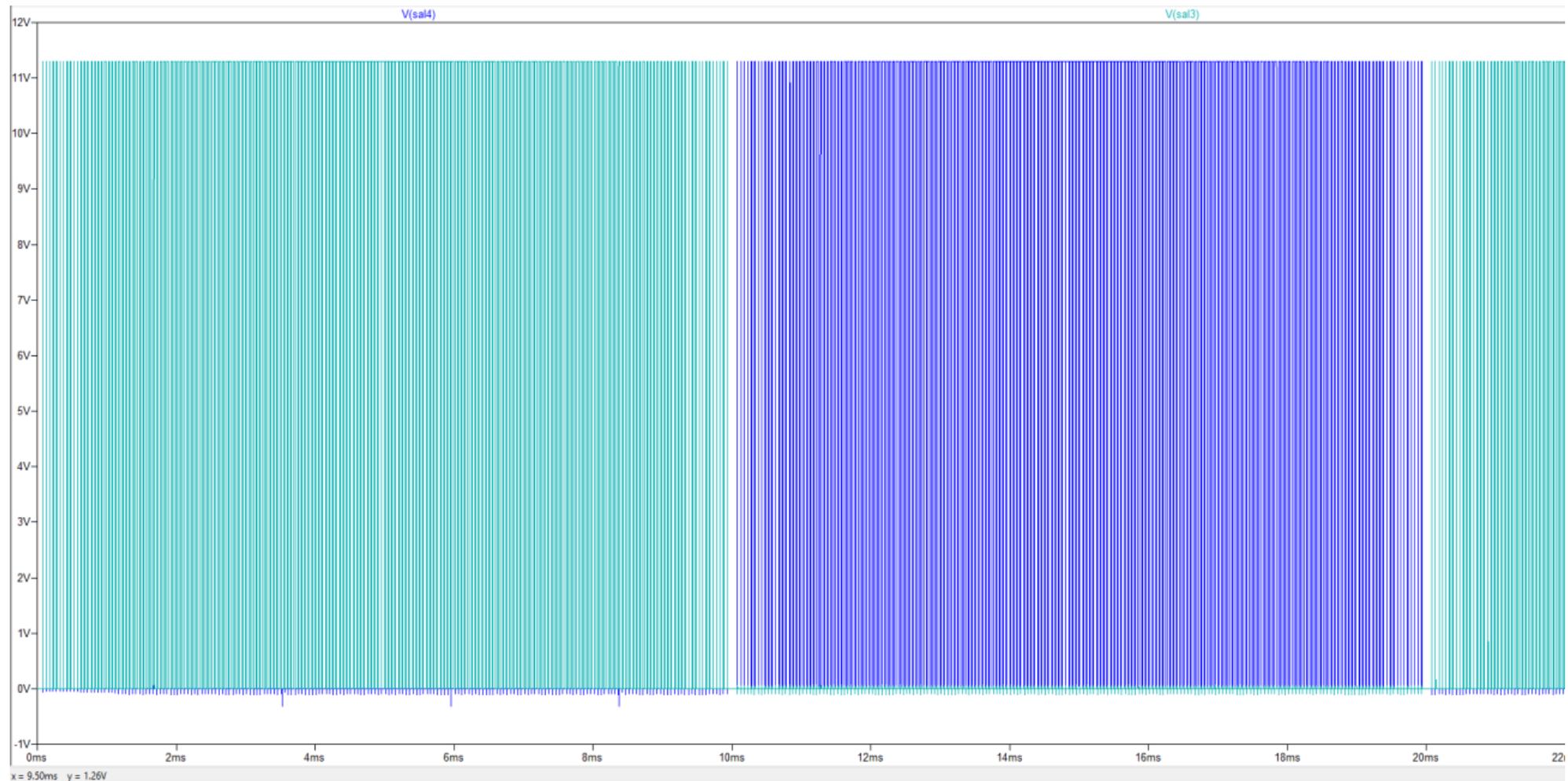


Figura 37: Voltaje de entrada a la puerta de los mosfets inferiores.

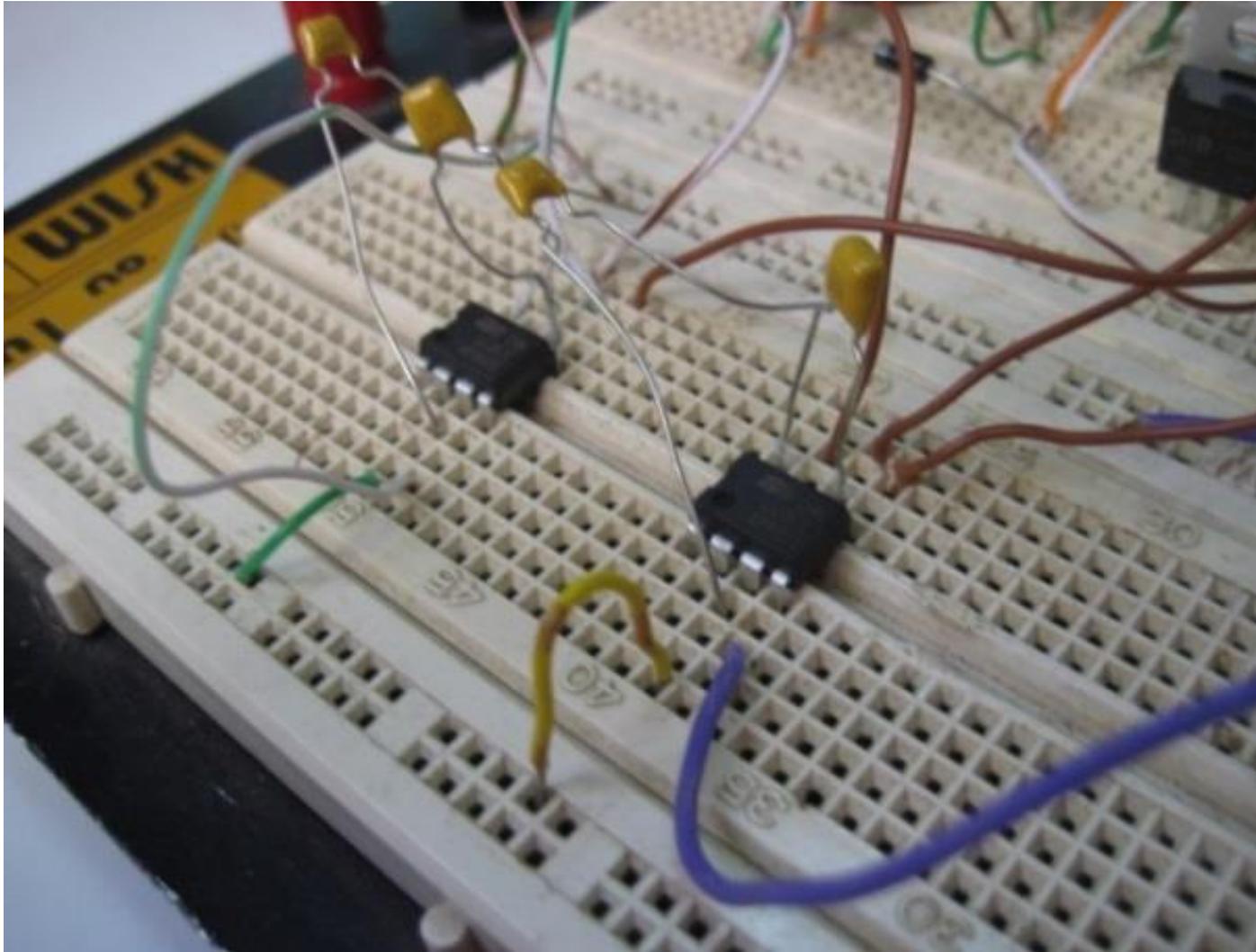


Figura 38: Fotografía del montaje del elevador de Spwm.

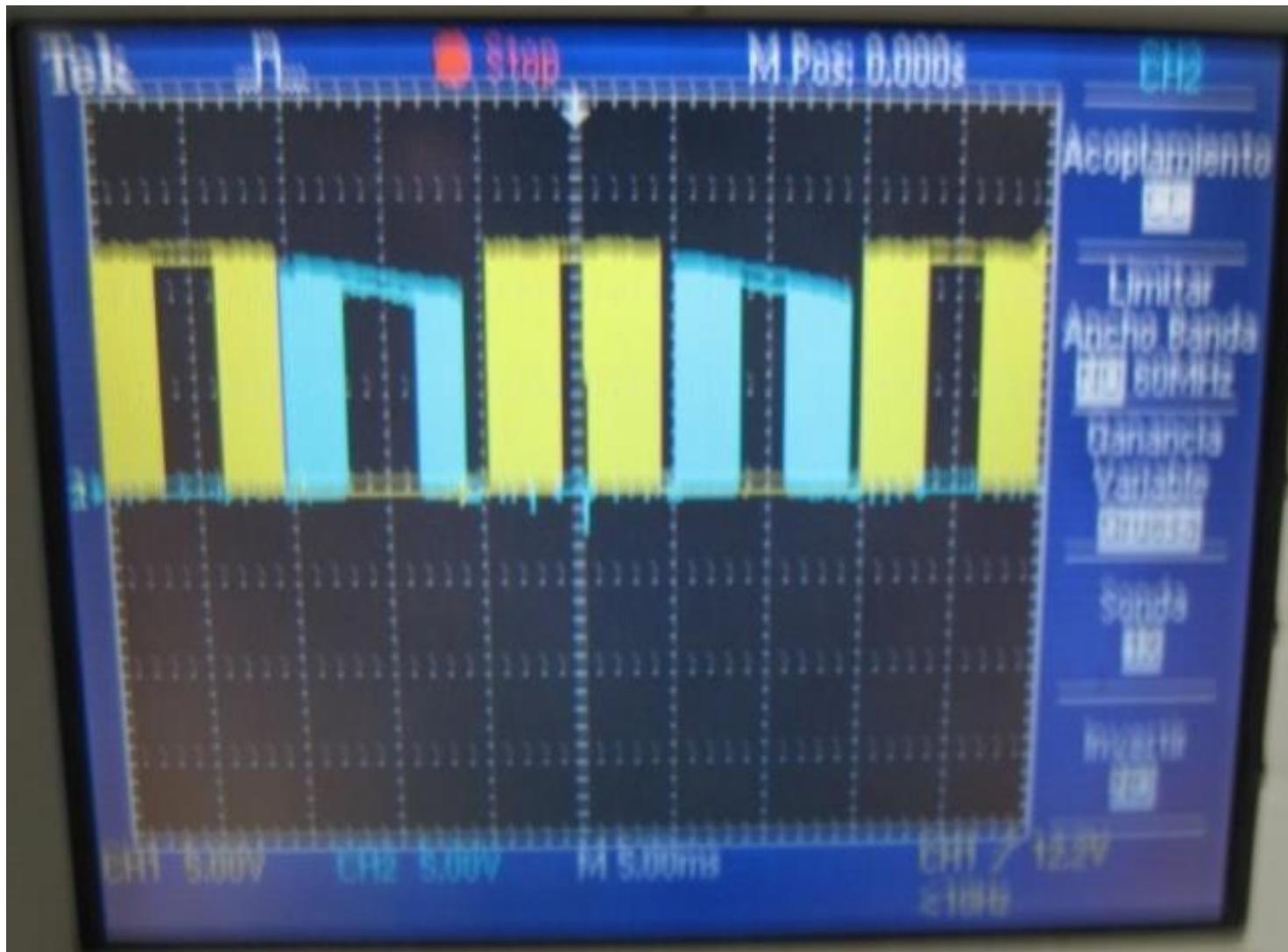


Figura 39: Fotografía de la salida del driver en el osciloscopio (Amarillo salida Bgate, azul salida Tgate)



Figura 40: Fotografía de las salidas del driver con carga conectada.(Amarillo salida Bgate, azul salida Tgate)

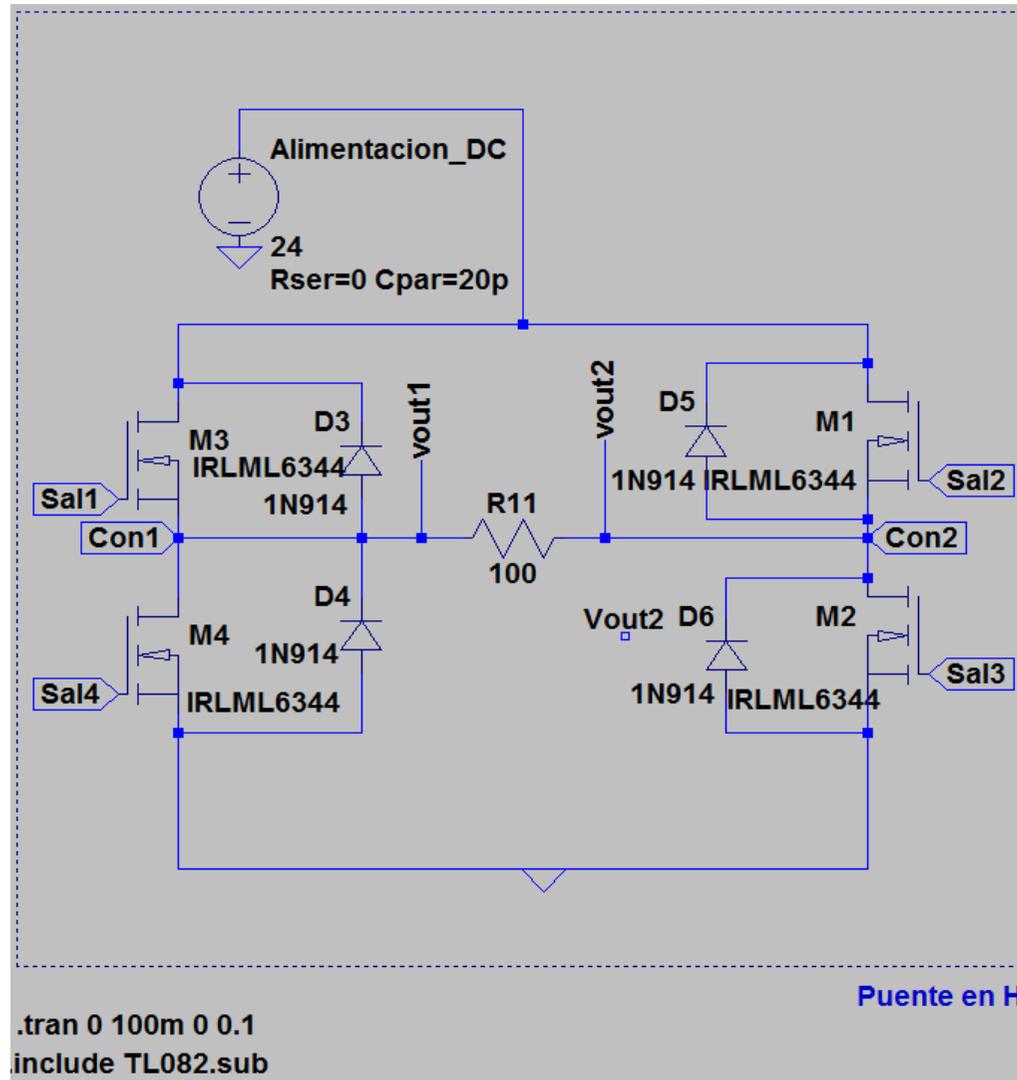


Figura 41: Esquema de la simulación del puente en H con carga resistiva.

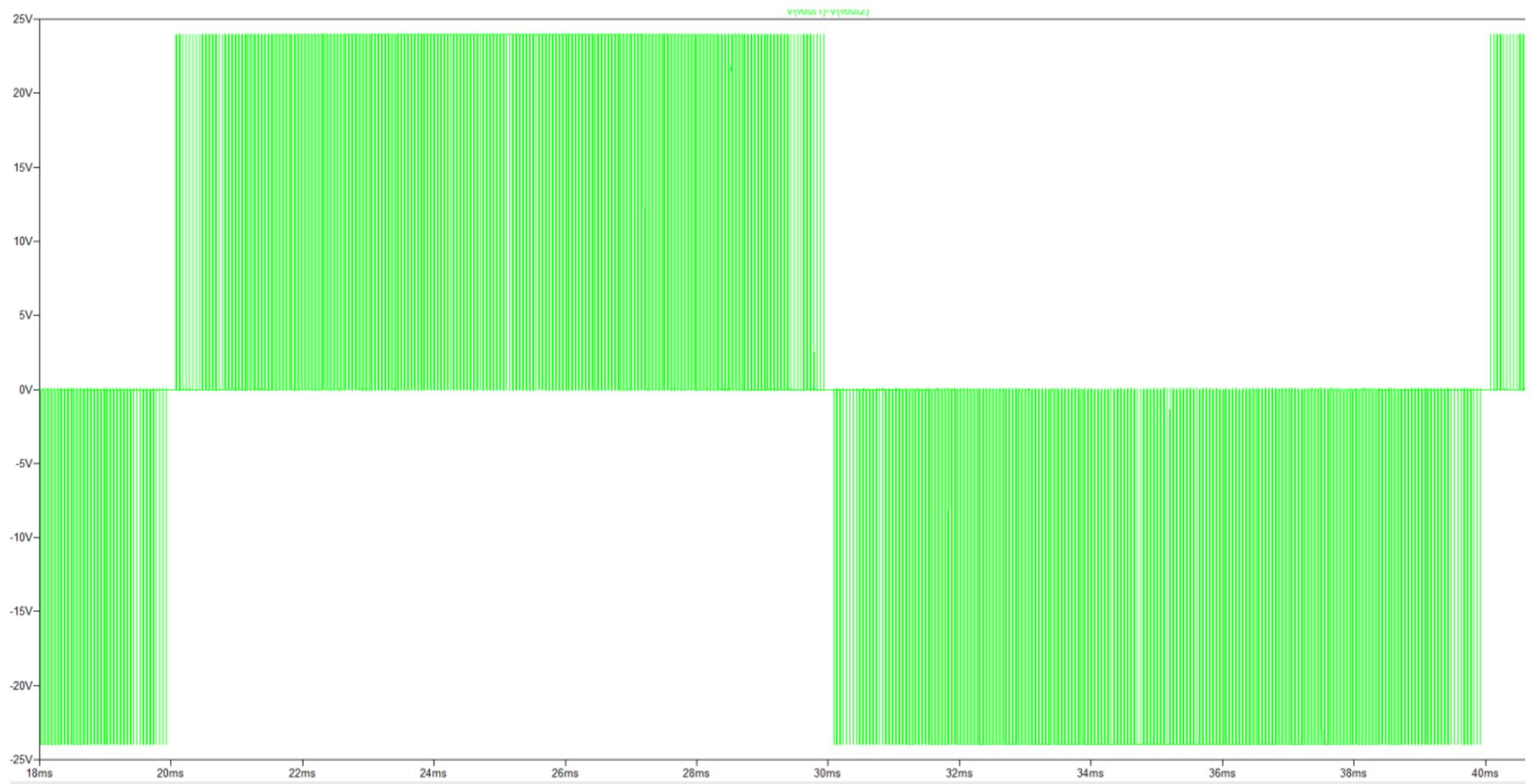


Figura 42: Salida del puente en H, simulación carga resistiva.

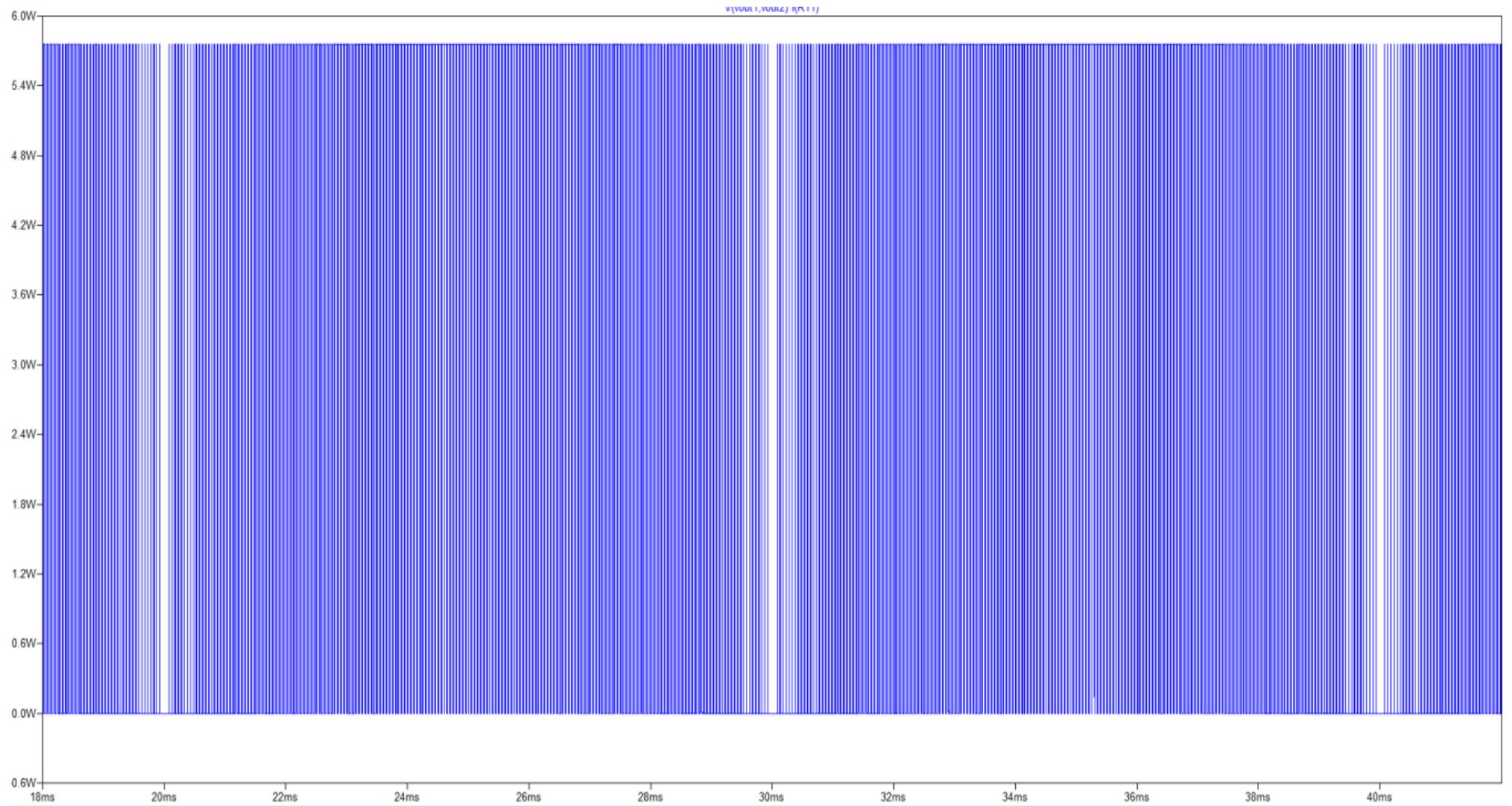


Figura 43: Potencia de la carga resistiva, Puente en H.

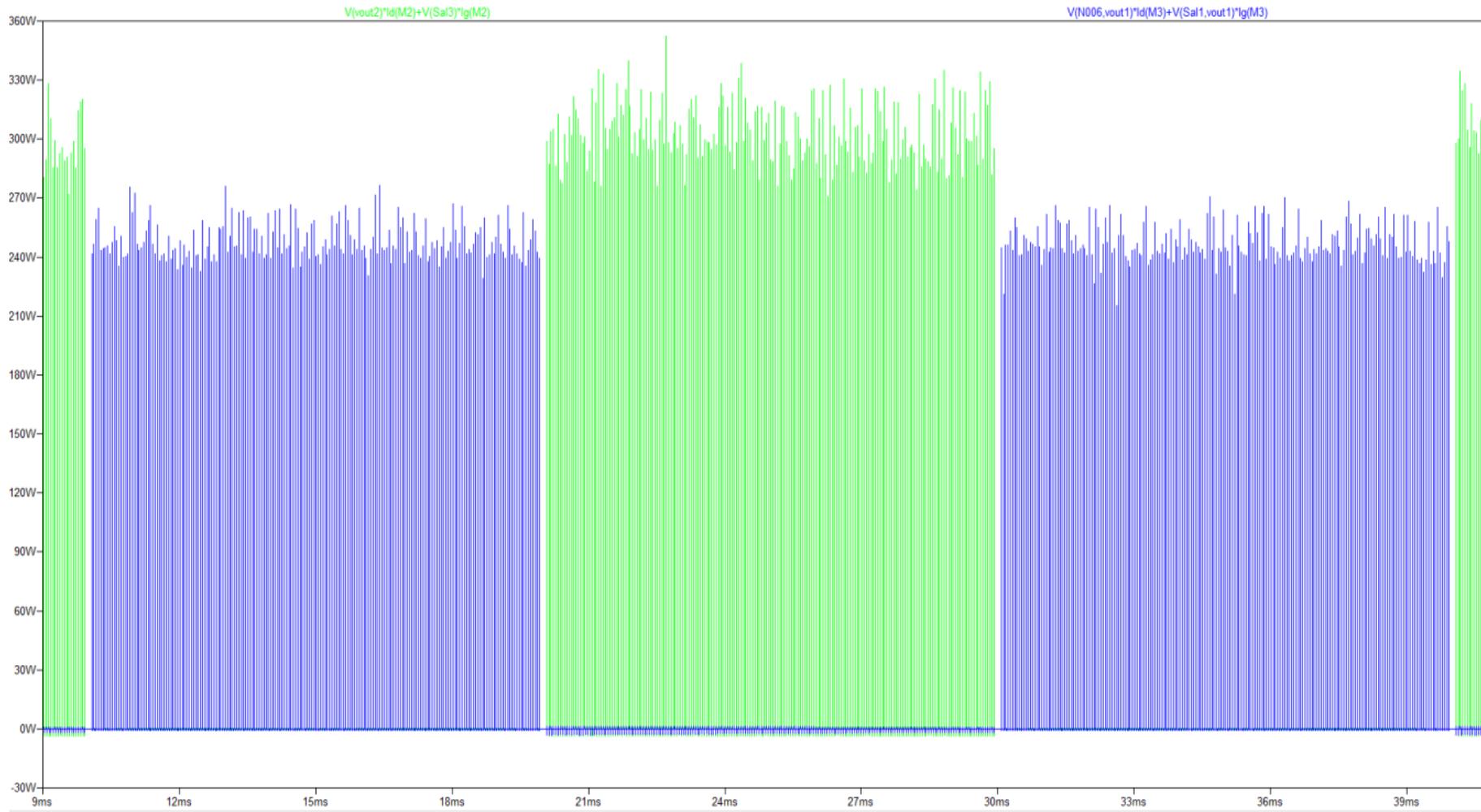


Figura 45: Potencia de los mosfet, Puente en H, carga resistiva.

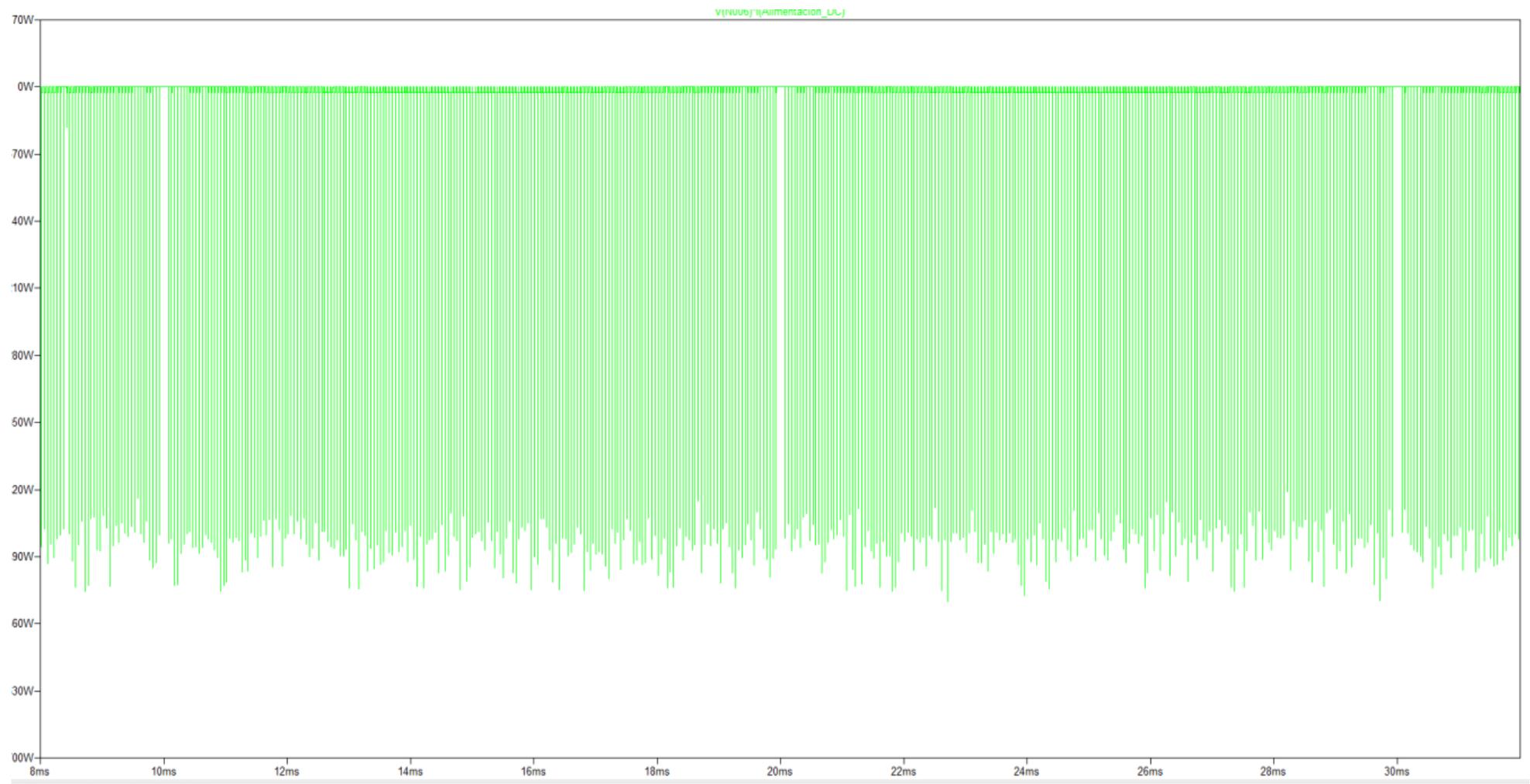


Figura 46: Potencia de la fuente de alimentación, Puente en H carga resistiva.

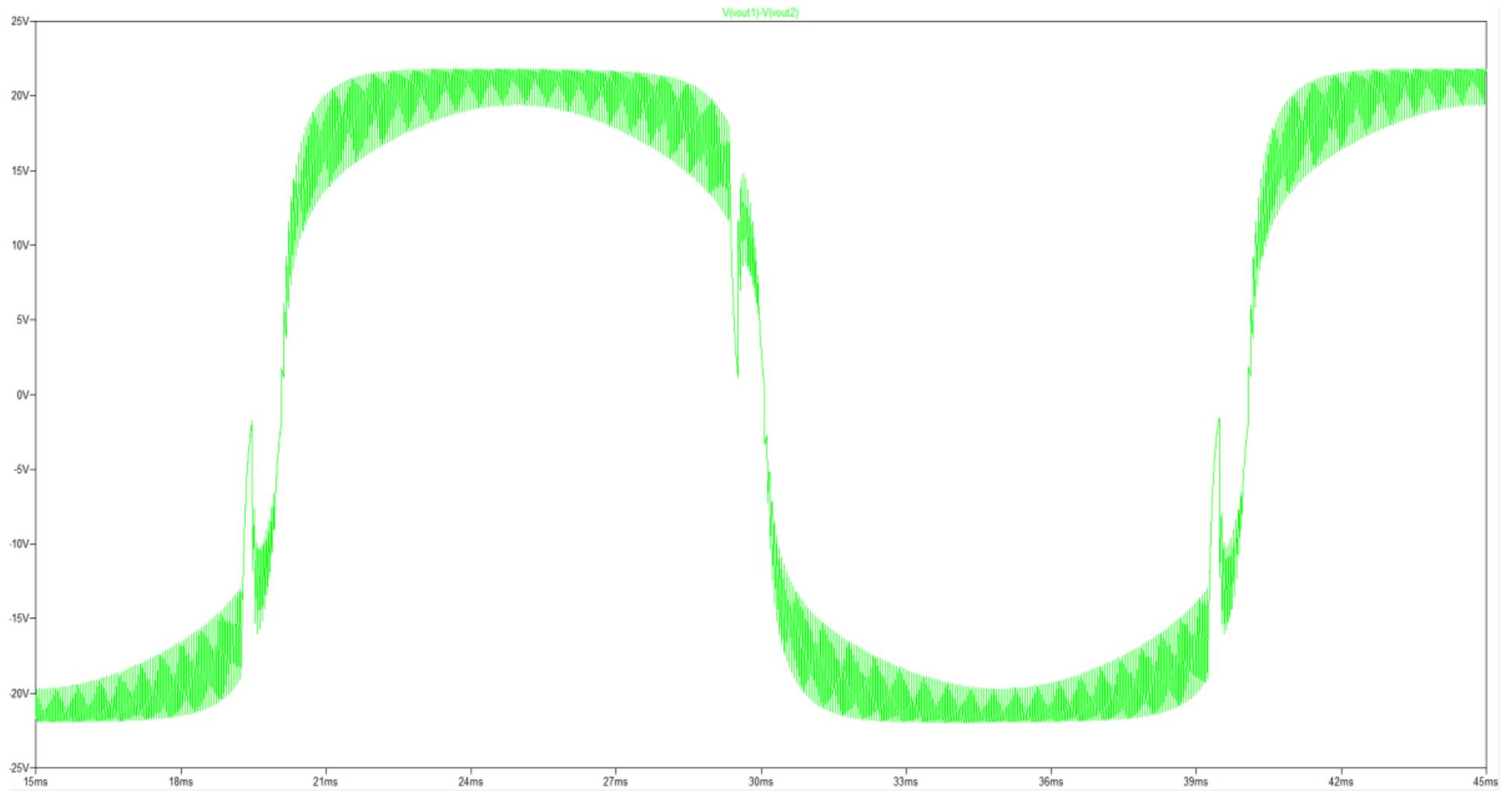


Figura 49: (a) Gráfica de la tensión de salida.

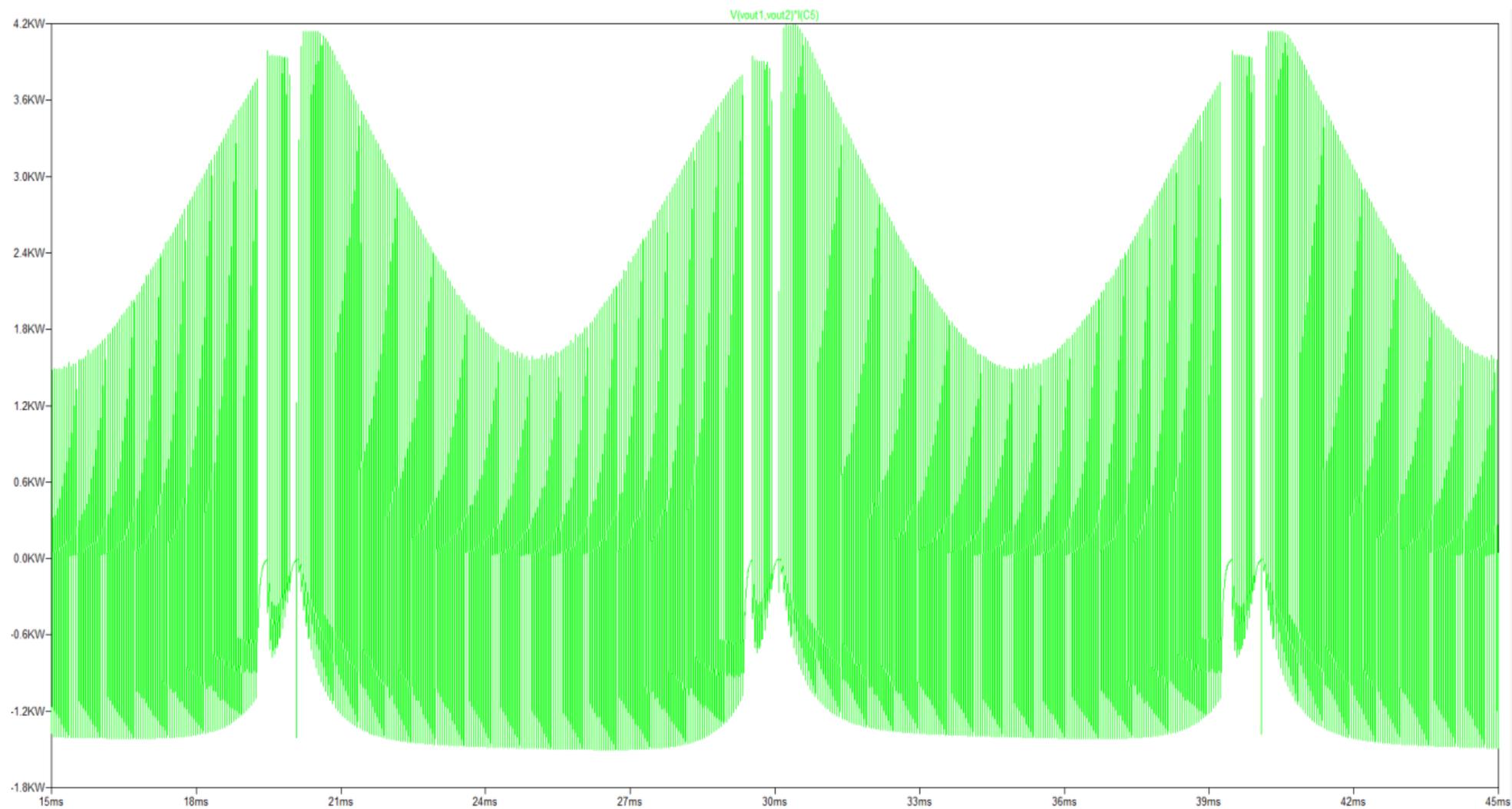


Figura 50: (a) Gráfica de la potencia del condensador

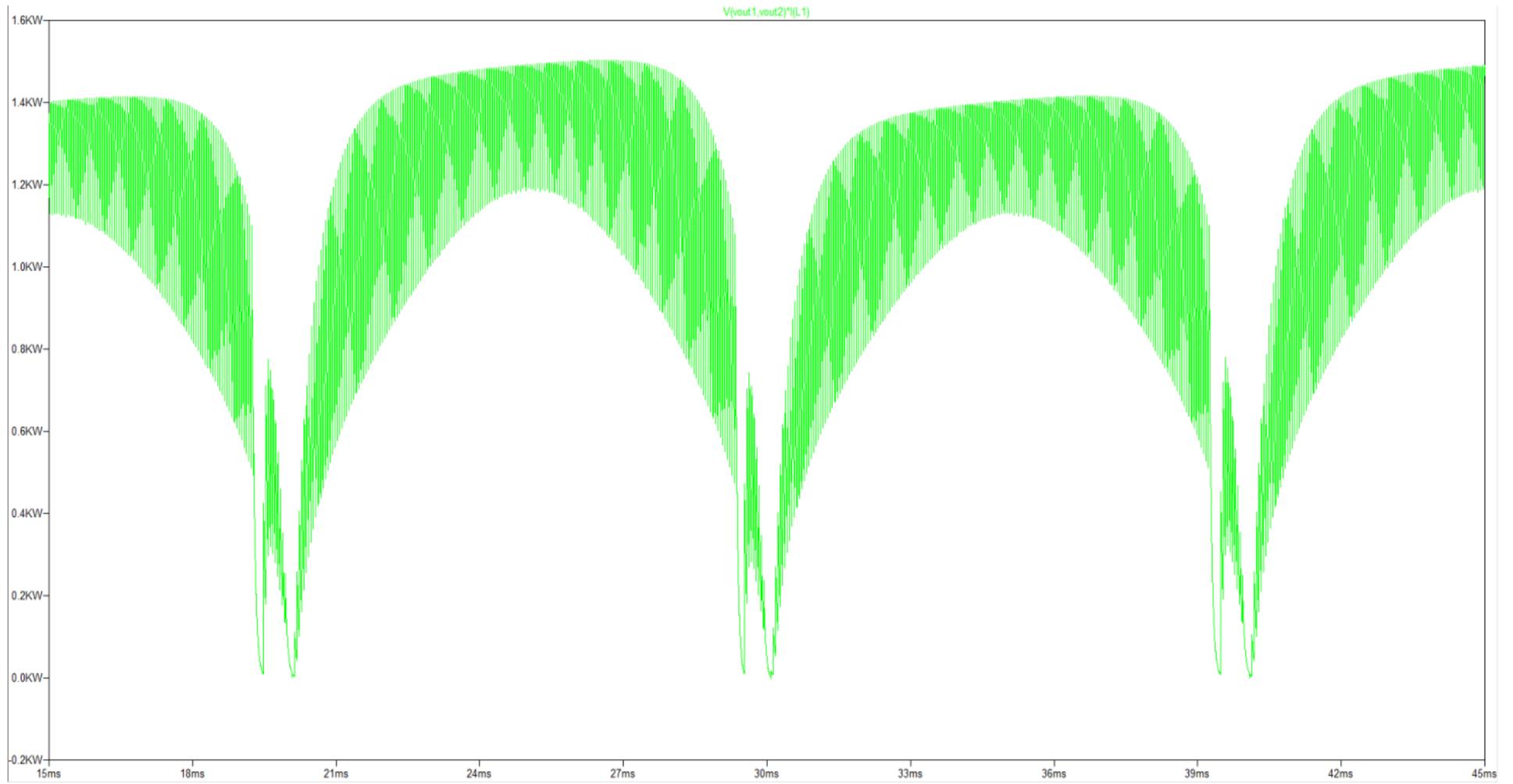


Figura 51: (a) Gráfica de la potencia de salida

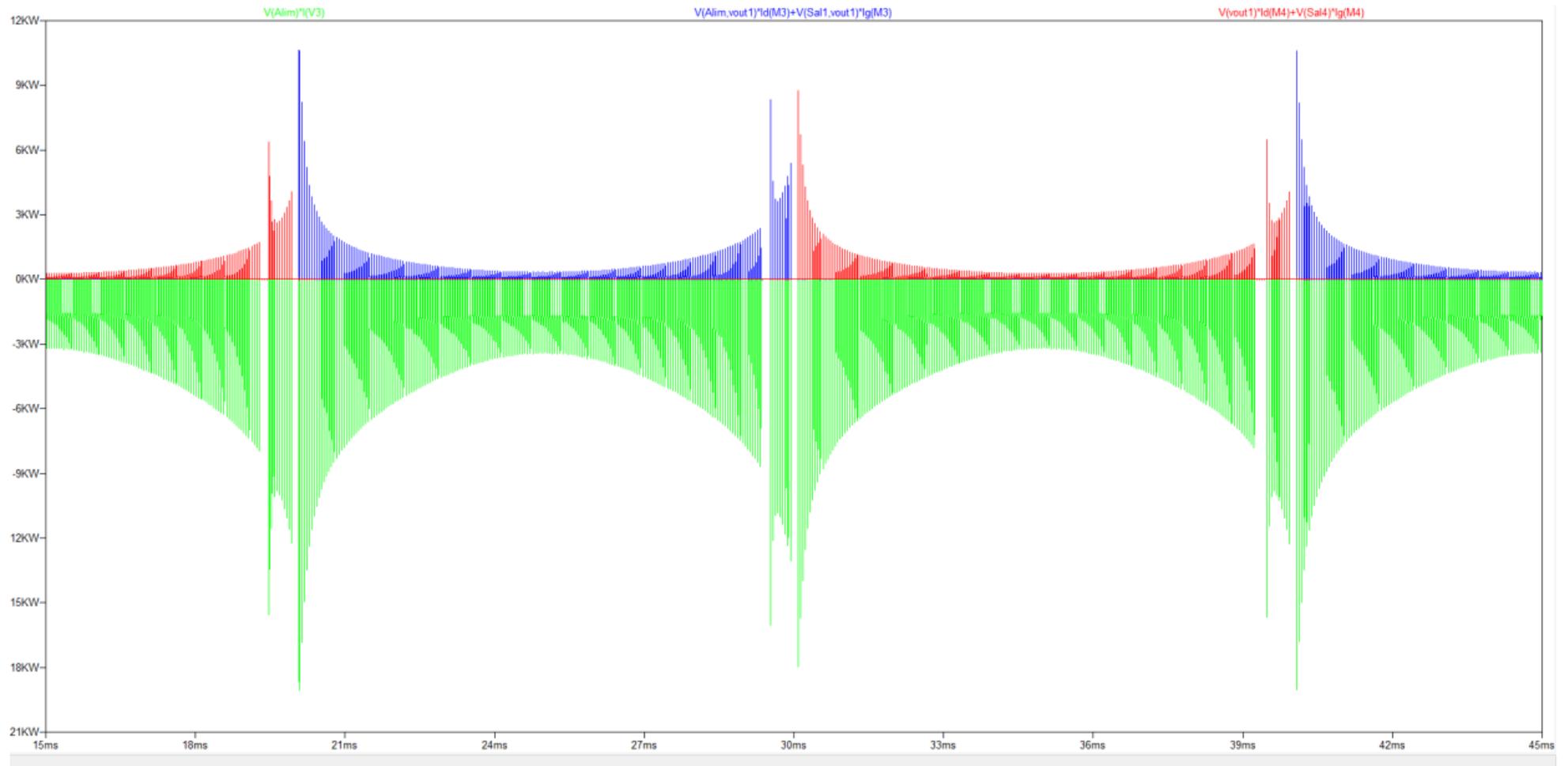


Figura 52: (a) Gráfica de la fuente de alimentación (verde) y la potencia disipada en los mosfet (azul y rojo)

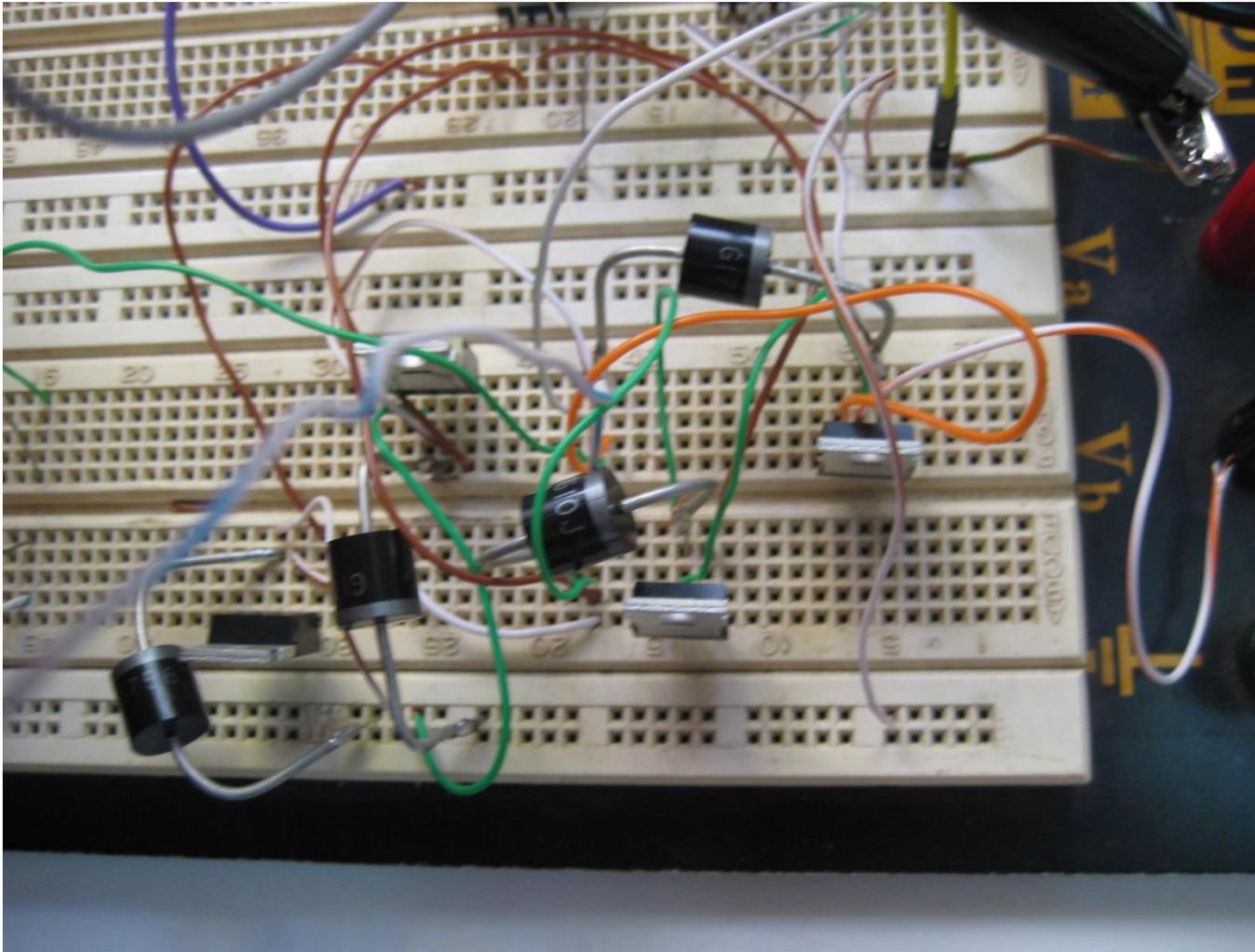


Figura 53: Fotografía del montaje del Puente en H sin carga.

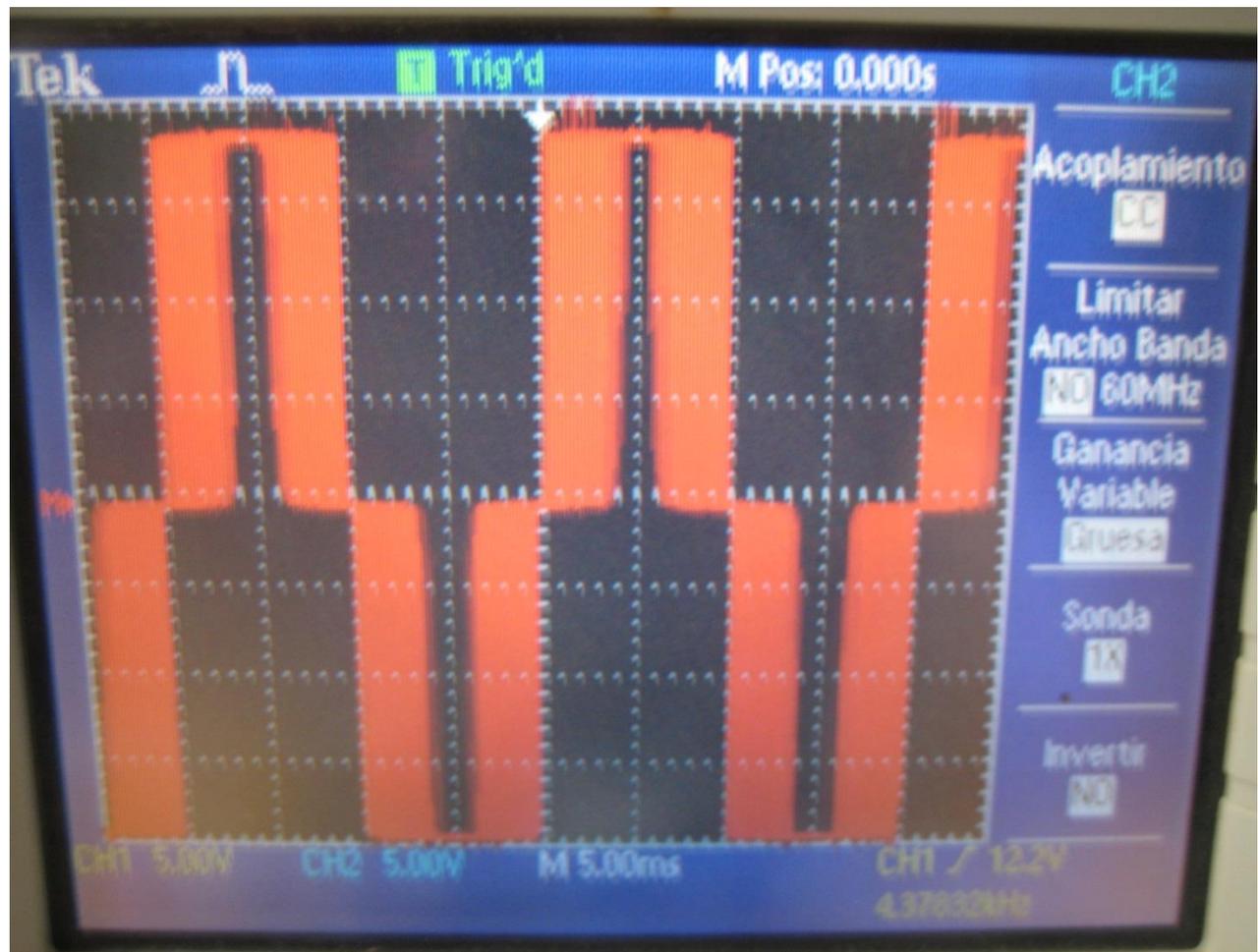


Figura 54: Fotografía de la tensión de salida con una carga resistiva.



Figura 55: Fotografía de la tensión en la carga con un filtro por condensador.

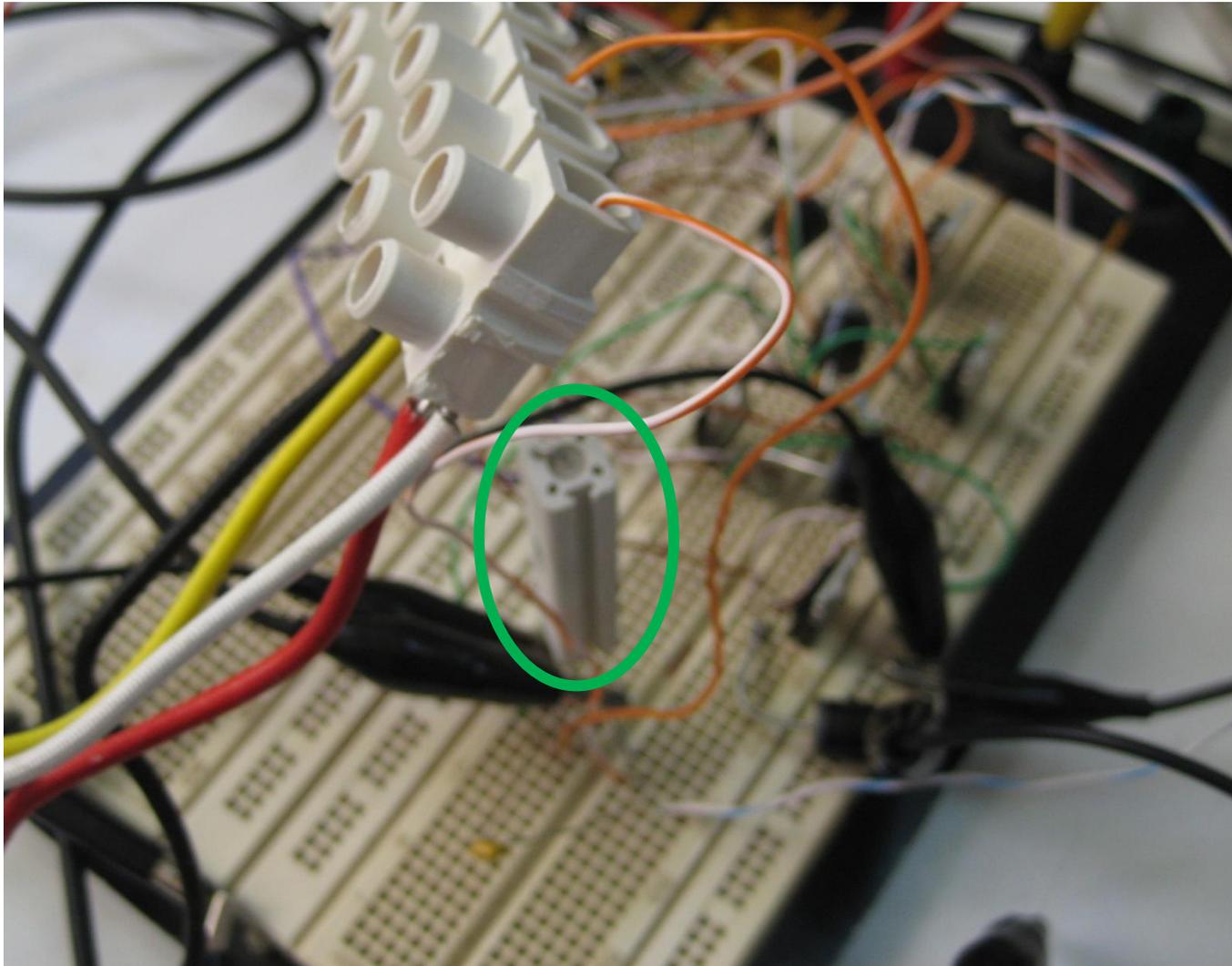


Figura 57: Fotografía que muestra la resistencia usada.

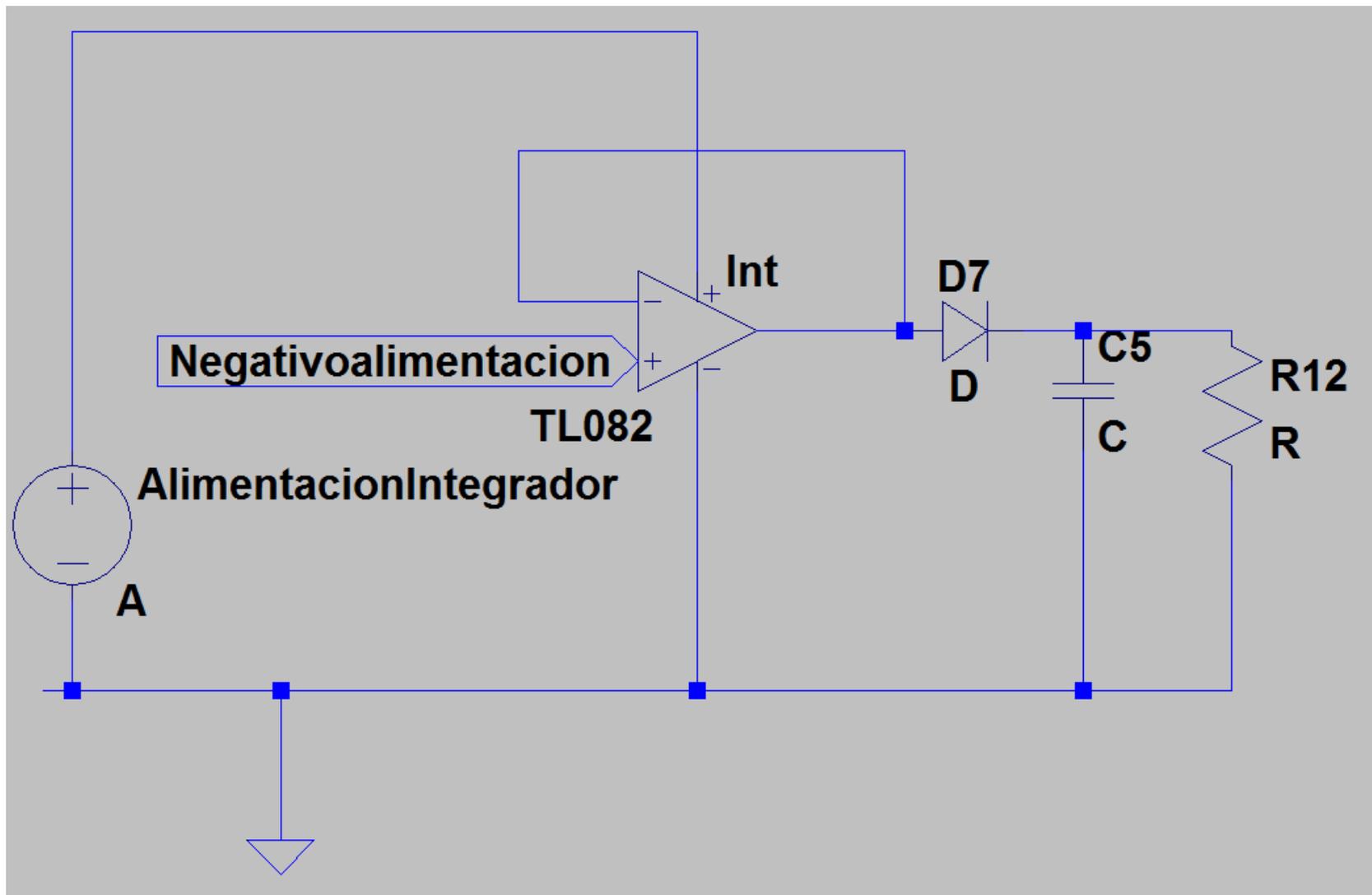


Figura 64: Esquema del integrador.

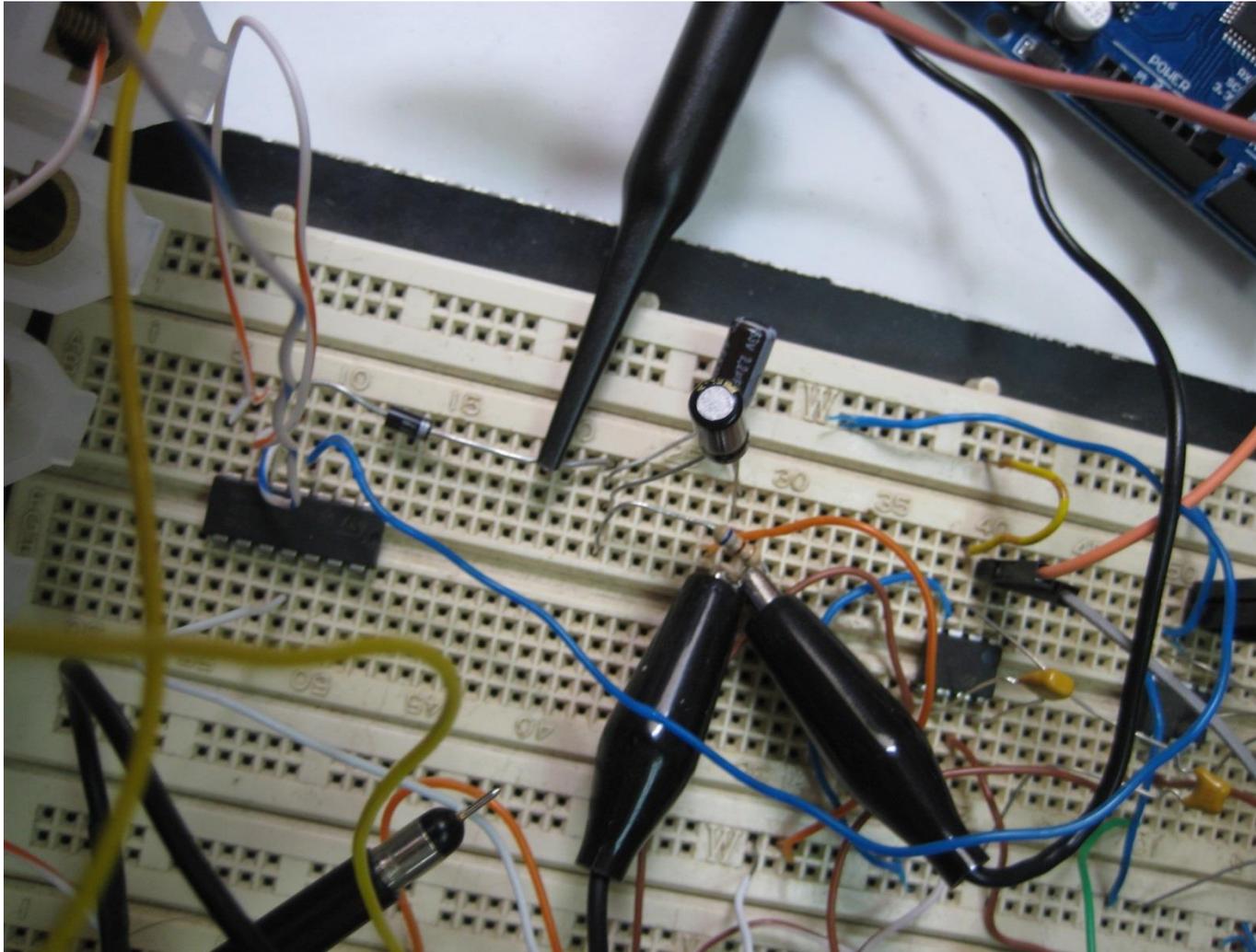


Figura 65: Fotografía del montaje de circuito integrador.

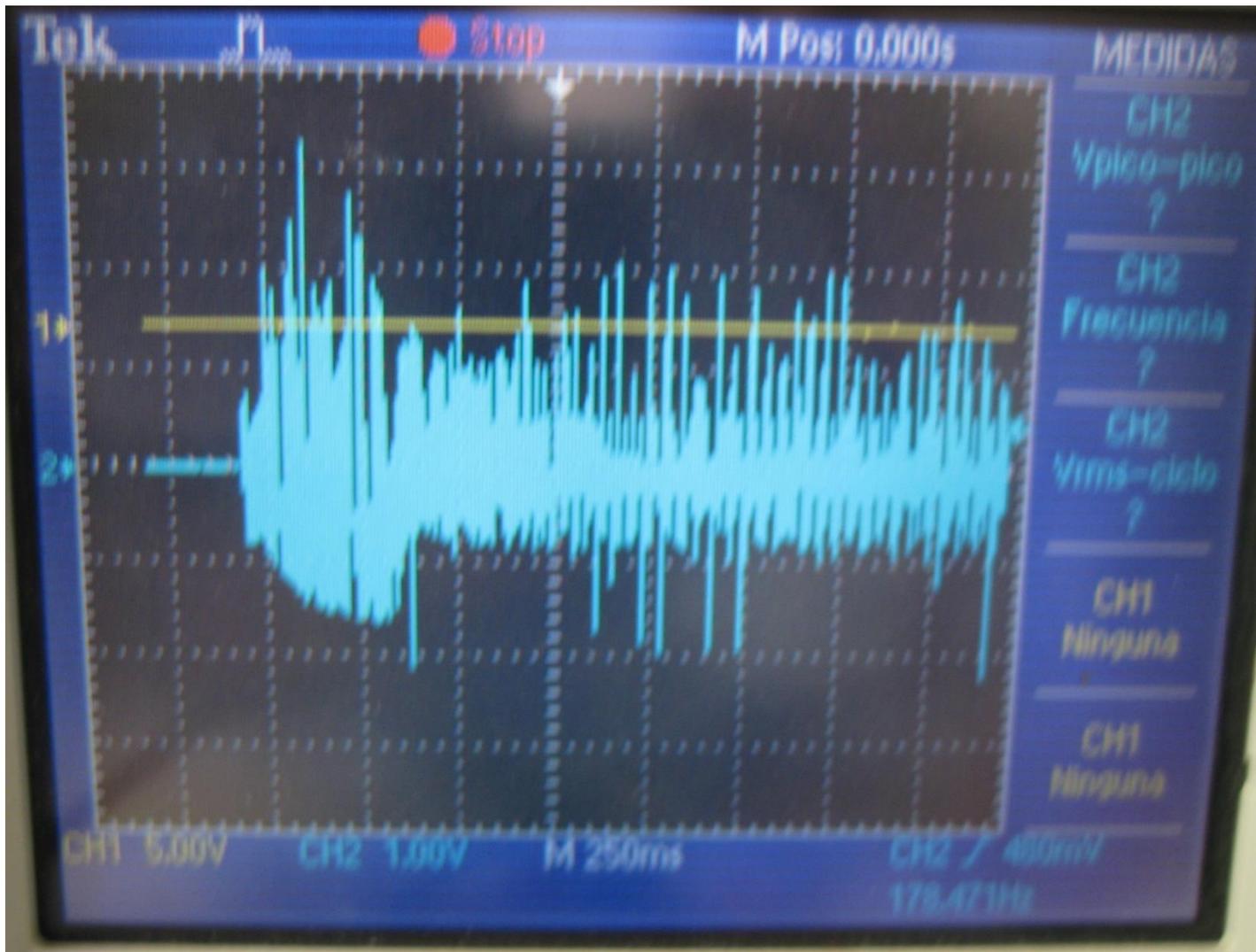


Figura 66: Fotografía que muestra la señal en el integrador y el pico de arranque.

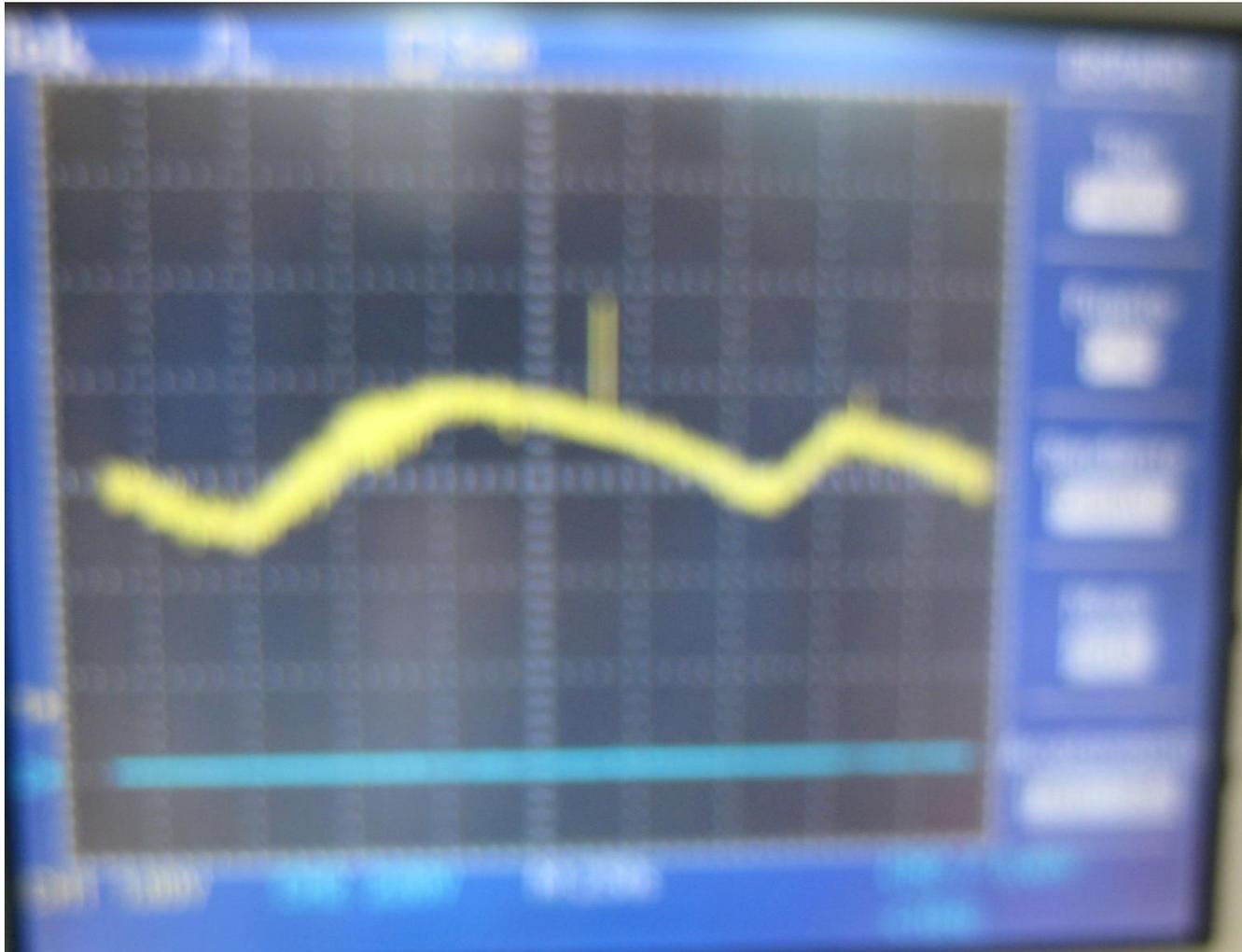


Figura 67: Fotografía que muestra la salida del integrador en más de 10 segundos.

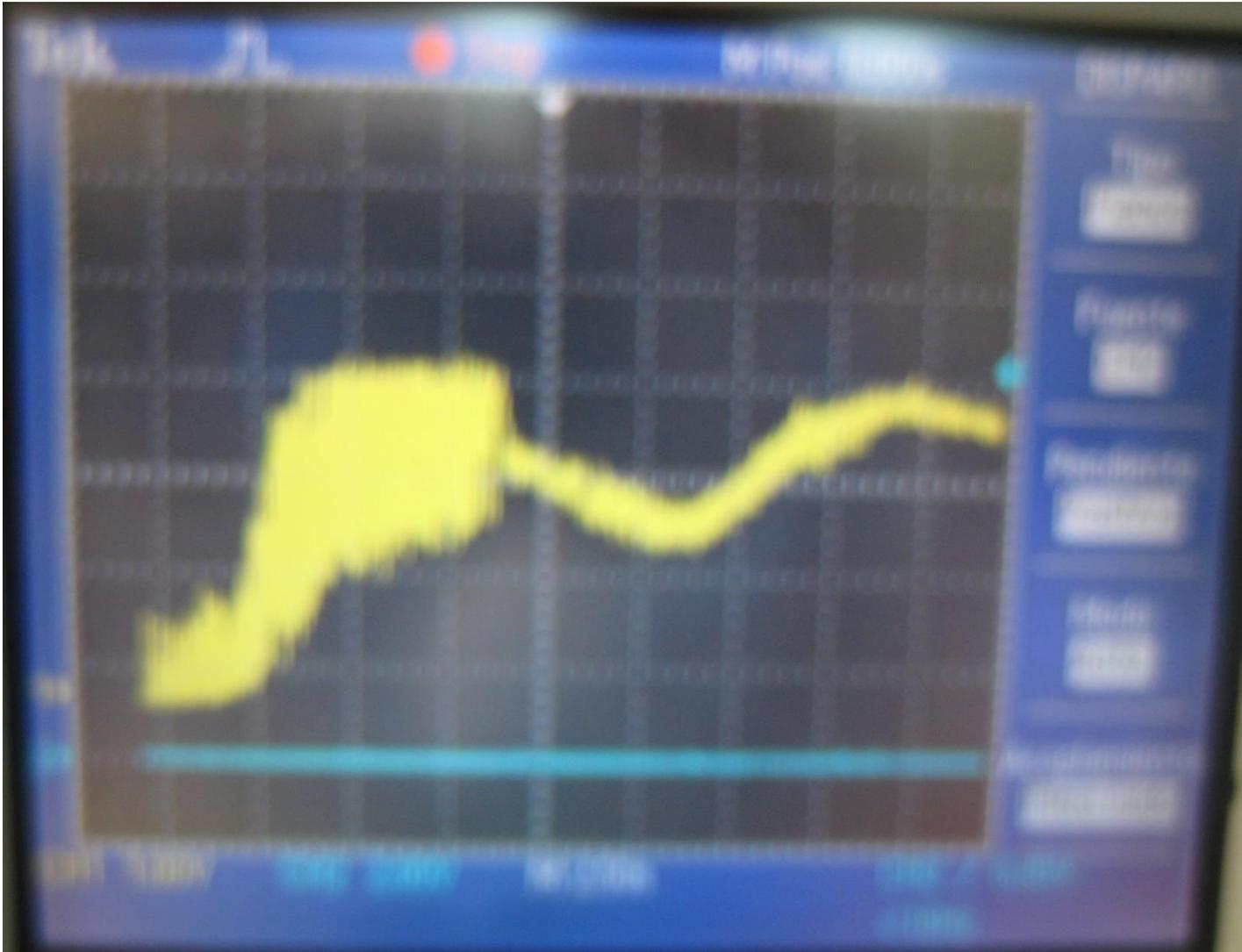


Figura 68: Fotografía que muestra la salida del integrador mientras se resetea el Arduino.

HOJAS DE DATOS

DATASHEET ATMEL328P (PORTADA Y CAPITULO 16.11)



ATmega48A/PA/88A/PA/168A/PA/328/P

**ATMEL 8-BIT MICROCONTROLLER WITH 4/8/16/32KBYTES
IN-SYSTEM PROGRAMMABLE FLASH**

DATASHEET

Features

- High Performance, Low Power Atmel® AVR® 8-Bit Microcontroller Family
- Advanced RISC Architecture
 - 131 Powerful Instructions – Most Single Clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 20 MIPS Throughput at 20MHz
 - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
 - 4/8/16/32KBytes of In-System Self-Programmable Flash program memory
 - 256/512/512/1KBytes EEPROM
 - 512/1K/1K/2KBytes Internal SRAM
 - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
 - Data retention: 20 years at 85°C/100 years at 25°C⁽¹⁾
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - Programming Lock for Software Security
- Atmel® QTouch® library support
 - Capacitive touch buttons, sliders and wheels
 - QTouch and QMatrix® acquisition
 - Up to 64 sense channels
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
 - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Six PWM Channels
 - 8-channel 10-bit ADC in TQFP and QFN/MLF package
 - Temperature Measurement
 - 6-channel 10-bit ADC in PDIP Package
 - Temperature Measurement
 - Programmable Serial USART
 - Master/Slave SPI Serial Interface
 - Byte-oriented 2-wire Serial Interface (Philips I²C compatible)
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
 - Interrupt and Wake-up on Pin Change

Atmel-4271J-AVR-ATmega-Datasheet_110216

16.11 Register Description

16.11.1 TCCR1A – Timer/Counter1 Control Register A

Bit	7	6	5	4	3	2	1	0	
(O/O)	COM1A1	COM1A0	COM1B1	COM1B0	–	–	WGM11	WGM10	TCCR1A
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- Bit 7:6 – COM1A1:0: Compare Output Mode for Channel A
- Bit 5:4 – COM1B1:0: Compare Output Mode for Channel B

The COM1A1:0 and COM1B1:0 control the Output Compare pins (OC1A and OC1B respectively) behavior. If one or both of the COM1A1:0 bits are written to one, the OC1A output overrides the normal port functionality of the I/O pin it is connected to. If one or both of the COM1B1:0 bits are written to one, the OC1B output overrides the normal port functionality of the I/O pin it is connected to. However, note that the Data Direction Register (DDR) bit corresponding to the OC1A or OC1B pin must be set in order to enable the output driver.

When the OC1A or OC1B is connected to the pin, the function of the COM1x1:0 bits is dependent of the WGM13:0 bits setting. Table 16-1 shows the COM1x1:0 bit functionality when the WGM13:0 bits are set to a Normal or a CTC mode (non-PWM).

Table 16-1. Compare Output Mode, non-PWM

COM1A1/COM1B1	COM1A0/COM1B0	Description
0	0	Normal port operation, OC1A/OC1B disconnected.
0	1	Toggle OC1A/OC1B on Compare Match.
1	0	Clear OC1A/OC1B on Compare Match (Set output to low level).
1	1	Set OC1A/OC1B on Compare Match (Set output to high level).

Table 16-2 shows the COM1x1:0 bit functionality when the WGM13:0 bits are set to the fast PWM mode.

Table 16-2. Compare Output Mode, Fast PWM⁽¹⁾

COM1A1/COM1B1	COM1A0/COM1B0	Description
0	0	Normal port operation, OC1A/OC1B disconnected.
0	1	WGM13:0 = 14 or 15: Toggle OC1A on Compare Match, OC1B disconnected (normal port operation). For all other WGM1 settings, normal port operation, OC1A/OC1B disconnected.
1	0	Clear OC1A/OC1B on Compare Match, set OC1A/OC1B at BOTTOM (non-inverting mode)
1	1	Set OC1A/OC1B on Compare Match, clear OC1A/OC1B at BOTTOM (inverting mode)

Note: 1. A special case occurs when OCR1A/OCR1B equals TOP and COM1A1/COM1B1 is set. In this case the compare match is ignored, but the set or clear is done at BOTTOM. See "Fast PWM Mode" on page 123 for more details.

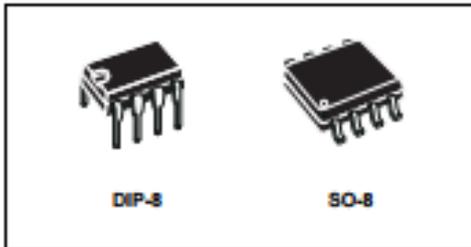
DATASHEET L6387E (PORTADA Y DIAGRAMA DE BLOQUES)



L6387E

High voltage high and low-side driver

Datasheet - production data



DIP-8

SO-8

Description

The L6387E is a simple and compact high voltage gate driver, manufactured with the BCD™ "offline" technology, and able to drive a half-bridge of power MOSFET or IGBT devices. The high-side (floating) section is enabled to work with voltage rail up to 600 V. Both device outputs can independently sink and source 650 mA and 400 mA respectively and cannot be simultaneously driven high thanks to an integrated interlocking function.

The L6387E device provides two input pins and two output pins and guarantees the outputs toggle in phase with inputs. The logic inputs are CMOS/TTL compatible to ease the interfacing with controlling devices.

The L6387E features the UVLO protection on the V_{CC} supply voltage and integrates the bootstrap diode, allowing a more compact and reliable solution.

The device is available in a DIP-8 tube and SO-8 tube and tape and reel packaging options.

Features

- High voltage rail up to 600 V
- dV/dt immunity ± 50 V/nsec in full temperature range
- Driver current capability
 - 400 mA source
 - 650 mA sink
- Switching times 50/30 nsec rise/fall with 1 nF load
- CMOS/TTL Schmitt trigger inputs with hysteresis and pull-down
- Internal bootstrap diode
- Outputs in phase with inputs
- Interlocking function

Applications

- Home appliances
- Motor drivers
 - DC, AC, PMDC and PMAC motors
- Lighting applications
- Industrial applications and drives
- Induction heating
- HVAC
- Factory automation
- Power supply systems

October 2015

DcdID13990 Rev 4

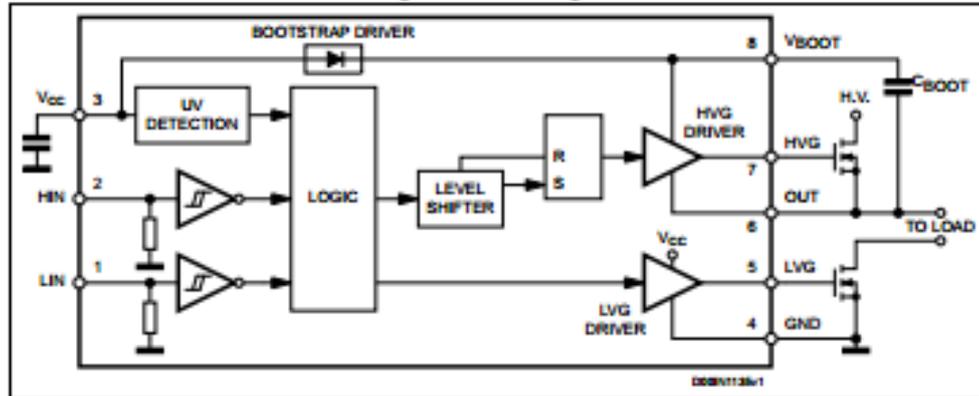
1/17

This is information on a product in full production.

www.st.com

1 Block diagram

Figure 1. Block diagram



DATASHEET 9NB60 (PORTADA)



STP9NB60 STP9NB60FP

N - CHANNEL 600V - 0.7Ω - 9A TO-220/TO220FP PowerMESH™ MOSFET

TYPE	V _{DS}	R _{DS(on)}	I _D
STP9NB60	600 V	< 0.8 Ω	9.0 A
STP9NB60FP	600 V	< 0.8 Ω	9.0 A

- TYPICAL R_{DS(on)} = 0.7 Ω
- EXTREMELY HIGH dv/dt CAPABILITY
- 100% AVALANCHE TESTED
- VERY LOW INTRINSIC CAPACITANCES
- GATE CHARGE MINIMIZED

DESCRIPTION

Using the latest high voltage MESH OVERLAY™ process, STMicroelectronics has designed an advanced family of power MOSFETs with outstanding performances. The new patent pending strip layout coupled with the Company's proprietary edge termination structure, gives the lowest R_{DS(on)} per area, exceptional avalanche and dv/dt capabilities and unrivalled gate charge and switching characteristics.

APPLICATIONS

- HIGH CURRENT, HIGH SPEED SWITCHING
- SWITCH MODE POWER SUPPLIES (SMPS)
- DC-AC CONVERTERS FOR WELDING EQUIPMENT AND UNINTERRUPTIBLE POWER SUPPLIES AND MOTOR DRIVE

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value		Unit
		STP9NB60	STP9NB60FP	
V _{DS}	Drain-source Voltage (V _{GS} = 0)	600		V
V _{DGSM}	Drain-gate Voltage (R _{GS} = 20 kΩ)	600		V
V _{GS}	Gate-source Voltage	± 30		V
I _D	Drain Current (continuous) at T _C = 25 °C	9.0	9.0(*)	A
I _D	Drain Current (continuous) at T _C = 100 °C	5.7	5.7(*)	A
I _{DM(*)}	Drain Current (pulsed)	36	36	A
P _{tot}	Total Dissipation at T _C = 25 °C	125	40	W
	Derating Factor	1.0	0.32	WPC
dv/dt(1)	Peak Diode Recovery voltage slope	4.5	4.5	V/ns
V _{ISO}	Insulation Withstand Voltage (DC)	—	2000	V
T _{stg}	Storage Temperature	-85 to 150		°C
T _J	Max. Operating Junction Temperature	150		°C

(*) Pulse width limited by safe operating area

(†) Limited only by maximum temperature allowed

(*) I_{DM} = 9A, dv/dt = 200 V/ns, V_{GS} = V_{DS(max)}, T_C = T_{amb}

