



# UNIVERSIDAD DE LA RIOJA

## TRABAJO FIN DE ESTUDIOS

Título

Fuente de alimentación programable de CC de alta eficiencia

Autor/es

ALBERTO OTERO OLAVARRIETA

Director/es

PEDRO MARÍA LARA SANTILLÁN

Facultad

Escuela Técnica Superior de Ingeniería Industrial

Titulación

Grado en Ingeniería Electrónica Industrial y Automática

Departamento

INGENIERÍA ELÉCTRICA

Curso académico

2018-19



***Fuente de alimentación programable de CC de alta eficiencia***, de ALBERTO  
OTERO OLAVARRIETA

(publicada por la Universidad de La Rioja) se difunde bajo una Licencia Creative  
Commons Reconocimiento-NoComercial-SinObraDerivada 3.0 Unported.

Permisos que vayan más allá de lo cubierto por esta licencia pueden solicitarse a los  
titulares del copyright.

© El autor, 2019

© Universidad de La Rioja, 2019

[publicaciones.unirioja.es](http://publicaciones.unirioja.es)

E-mail: [publicaciones@unirioja.es](mailto:publicaciones@unirioja.es)



**UNIVERSIDAD  
DE LA RIOJA**

**ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL**

## **TRABAJO DE FIN DE GRADO**

**TITULACIÓN: Grado en  
Ingeniería Electrónica Industrial y Automática**

**CURSO: 2018-2019**

**CONVOCATORIA: SEPTIEMBRE**

**TÍTULO:**

**Fuente de alimentación programable de  
CC de alta eficiencia**

**ESTUDIANTE: Alberto Otero Olavarrieta**

**TUTORES/AS: Pedro María Lara Santillán**

**DEPARTAMENTO: Ingeniería Eléctrica**





MEMORIA.....	5
ANEXO I: CÁLCULOS.....	129
ANEXO II: SOFTWARE UTILIZADO.....	151
ANEXO III: PRUEBAS.....	161
ANEXO IV: HOJAS DE DATOS.....	173
ANEXO V: CÓDIGO.....	185
PLANOS.....	271
PLIEGO DE PRESCRIPCIONES TÉCNICAS.....	335
MEDICIONES.....	347
PRESUPUESTO.....	363
Bibliografía.....	381



# MEMORIA



## Índice

Resumen.....	19
Abstract .....	19
1. Introducción .....	21
2. Antecedentes .....	22
2.1. Estructura de una fuente de alimentación conmutada .....	22
2.2. Filtrado EMI .....	23
2.2.1. Ruido en modo diferencial .....	23
2.2.2. Ruido en modo común .....	23
2.2.3. Filtro más extendido.....	23
2.3. El factor de potencia .....	24
2.3.1. Métodos de corrección f.d.p.....	27
2.3.2. Métodos de corrección f.d.p – Corrección Pasiva.....	27
2.3.3. Métodos de corrección f.d.p – Convertidores modernos.....	30
Convertidor elevador de tensión (Boost).....	30
Modos de operación.....	32
2.4. Convertidor DC - DC .....	33
2.4.1. Convertidor Flyback.....	33
2.4.2. Convertidor Medio puente (Half Bridge) .....	34
2.4.3. Convertidor Puente Completo (Full Bridge).....	34
2.4.4. ZVS (Zero Voltage Switching) .....	35
Operación y formas de onda .....	36
2.4.5. IGBT vs MOSFET .....	37
2.4.6. Rectificación Activa .....	38
2.4.7. Diodos vs Rectificador Síncrono.....	39
3. Requisitos de diseño .....	40
4. Descripción de soluciones e implementaciones .....	42
4.1. Panel frontal y botones de control.....	42
4.1.1. Etapa de salida .....	43
4.1.2. Etapa de salida – Conectores 4 mm .....	43
4.1.3. Botones de control de la fuente y antirrebote hardware.....	43
4.1.4. Circuito Impreso .....	47
4.1.5. Pantalla Nextion HMI .....	47
4.1.6. Interruptor AC .....	48

4.1.7.	Construcción.....	49
4.2.	Panel trasero .....	51
4.2.1.	Conector AC.....	51
4.2.2.	Fabricación .....	52
4.2.3.	Fuentes auxiliares.....	53
4.2.4.	Posibilidad de mejora.....	54
4.2.5.	Cuerpo de la fuente de alimentación.....	55
4.2.6.	Lámina inferior .....	55
4.2.7.	Cubierta superior.....	56
4.3.	Sistema de filtrado EMI y convertidor PFC.....	57
4.3.1.	Etapas de filtro EMI y arranque suave.....	57
4.3.2.	Elemento de protección contra sobrecorriente: Fusibles.....	58
4.3.3.	Elemento de protección contra sobretensiones: Varistor .....	58
4.3.4.	Filtro EMI .....	59
4.3.5.	Arranque suave .....	59
4.3.6.	Terminales de conexión de potencia .....	60
4.3.7.	Resultado del circuito impreso.....	61
4.3.8.	Rectificado y convertidor PFC .....	62
4.3.9.	Rectificador de onda completa .....	63
4.3.10.	Convertidor DC-DC elevador de tensión .....	65
4.3.11.	Bobina PFC.....	65
4.3.12.	MOSFET Boost PFC .....	66
4.3.13.	Diodo de rectificación Boost .....	66
4.3.14.	Condensadores de filtro de salida.....	67
4.3.15.	Diseño de PCB .....	67
4.3.16.	Fabricación PCB .....	68
4.3.17.	Controlador PFC .....	70
4.3.18.	Configuración de la frecuencia de conmutación.....	70
4.3.19.	Arranque suave .....	71
4.3.20.	Lazo de tensión.....	72
4.3.21.	Lazo de corriente.....	73
4.3.22.	Driver MOSFET .....	73
4.3.23.	Resultado PCB .....	74
4.3.24.	Montaje final .....	74
4.4.	Convertidor DC – DC Puente Completo ZVS.....	75

4.4.1.	Dimensionamiento puente completo .....	75
4.4.2.	MOSFET puente completo .....	76
4.4.3.	Diodo volante .....	76
4.4.4.	Diseño circuito de medida transformador de corriente .....	77
4.4.5.	Diseño transformador .....	78
4.4.6.	Alternativas transformador.....	79
4.4.7.	Terminales Faston 6.3mm.....	80
4.4.8.	Bobina de resonancia .....	80
4.4.9.	Red RCD contra picos de tensión y oscilaciones .....	81
4.4.10.	Resistencia de medida de corriente (Shunt) .....	81
4.4.11.	Rectificación síncrona en el secundario .....	82
4.4.12.	MOSFET rectificación síncrona.....	83
4.4.13.	Rectificación síncrona Autónoma (Self Driven).....	83
4.4.14.	Diseño etapa de filtrado en el secundario .....	83
4.4.15.	Construcción.....	84
4.5.	Drivers MOSFET Puente Completo (Primario) .....	86
4.5.1.	Drivers en el lado de alta tensión vs lado de baja tensión.....	86
4.5.2.	Drivers desarrollados .....	87
4.6.	Driver MOSFET Rectificación Síncrona (Secundario).....	89
4.7.	Regulación de temperatura y control de ventiladores .....	90
4.7.1.	Disipadores utilizados .....	90
4.7.2.	Sensores utilizados y acondicionamiento .....	91
4.7.3.	Localización de los sensores.....	92
4.7.4.	MOSFET de control de ventiladores.....	92
4.7.5.	Ventiladores utilizados.....	93
4.7.6.	Control de ventiladores mediante PWM .....	93
4.7.7.	Microcontrolador ATtiny 84.....	94
4.7.8.	Configuración y generación de señales PWM.....	94
4.7.9.	Estructura del programa .....	94
	Diagrama de flujo.....	95
4.7.10.	Circuito impreso .....	96
4.7.11.	Funcionamiento .....	96
4.7.12.	Consumos y eficiencia .....	98
4.8.	Placa controladora convertidor DC – DC.....	99
4.8.1.	dsPIC33EV256GM002.....	99

4.8.2.	Regulador de tensión .....	100
4.8.3.	Referencia de tensión .....	100
4.8.4.	Referencia utilizada: TL431A .....	101
4.8.5.	Medidas analógicas – Corriente en el primario .....	102
4.8.6.	Medidas analógicas – Tensión en el secundario .....	102
4.8.7.	Seguidor de tensión.....	102
4.8.8.	Medidas analógicas – Corriente en el secundario .....	103
4.8.9.	Generador de tensión negativa: TC7660.....	103
4.8.10.	Lectura de tensión en el shunt: Amplificador de instrumentación.....	104
4.8.11.	Pruebas con otros operacionales .....	105
	AD8230 de Analog Devices .....	105
	AD8422BRZ de Analog Devices .....	105
4.8.12.	Alternativa TC7660: ADP3605 .....	105
4.8.13.	Puerto I2C.....	106
4.8.14.	Conmutación relé de salida.....	106
4.8.15.	Estructura del programa .....	106
	Configuración – MCC.....	107
	ADC 12 bits .....	107
	Módulo y Salidas PWM.....	108
	Configuración I2C.....	109
	Rutinas de interrupción – Timer 1 .....	109
	Rutinas de interrupción – Cambio de estado RB8 .....	110
	Programa principal .....	110
4.8.16.	Lazo de control por pico de corriente en el primario.....	111
4.8.17.	Implementación controlador 2P2Z .....	111
4.8.18.	Implementación con librería SMPS Control Library de Microchip.....	112
4.8.19.	DAC y comparador digitales .....	113
4.8.20.	Anti Wind-Up y saturaciones PWM.....	113
4.8.21.	Construcción.....	113
4.9.	Placa de gestión del panel frontal .....	114
4.9.1.	Atmega328P .....	115
4.9.2.	Convertor UART – USB: CH340G.....	115
4.9.3.	Alternativas CH340G .....	116
4.9.4.	Regulador de tensión .....	117
4.9.5.	Estructura del programa .....	117

4.9.6.	Configuración y recursos utilizados.....	118
4.9.7.	Programa principal.....	120
4.9.8.	Puerto I2C.....	121
4.9.9.	Construcción.....	122
4.10.	Control bajo PC.....	123
4.10.1.	Control bajo PC - Bloque Gestión Puerto USB.....	126
5.	Conclusiones.....	126



## Índice de ilustraciones

Ilustración 1 – Modelo 3D Fuente completa: Vista exterior .....	21
Ilustración 2 - Esquema fuente de alimentación conmutada .....	22
Ilustración 3 - Esquema típico filtro EMI. Fuente: Practical Design of a Passive EMI Filter for Reduction of EMI Generation.....	23
Ilustración 4 - Arriba: Tensión. Abajo: Corriente. Se observa que la tensión solo cae en el centro de la señal. Fuente: powerelectrontips.com .....	24
Ilustración 5 - Carga armónica en la línea sin corrección f.d.p. Fuente: powerelectrontips.com .....	25
Ilustración 6 - Izquierda: Forma de onda sin distorsión. Derecha: El nivel de armónicos es el de la frecuencia fundamental .....	26
Ilustración 7 - Esquema típico de un convertidor Boost PFC. Fuente: powerelectrontips.com .....	26
Ilustración 8 - Formas de onda de corriente con diferentes métodos de corrección. Fuente: powerelectrontips.com .....	26
Ilustración 9 - Estructura fuente de alimentación Xantrex 300V4A. Fuente: EEVblog .....	27
Ilustración 10 - Corrección pasiva f.d.p. con bobina en el lado de alterna. Fuente: sound-au.com.....	28
Ilustración 11 - Corrección pasiva f.d.p. con bobina en el lado de continua. Fuente: sound-au.com.....	28
Ilustración 12 - Comparación corriente en la bobina en lado AC (Rojo) y en lado DC (Verde). Fuente: sound-au.com .....	28
Ilustración 13 - Pico de tensión en el encendido .....	29
Ilustración 14 - Fuente de alimentación de PC con corrección pasiva del f.d.p. Fuente: sound-au.com.....	30
Ilustración 15- Esquema básico Boost PFC. Fuente: Infineon .....	30
Ilustración 16 – Estructura fuente de alimentación Manson SPS-9400. Fuente: EEVblog .....	31
Ilustración 17 - Modos de operación convertidor Boost PFC. Fuente: PFC boost converter design guide. Infineon.....	32
Ilustración 18 - Esquema básico convertidor Flyback. Fuente: Topologies for switched mode power supplies. STMicroelectronics .....	33
Ilustración 19 - Esquema básico convertidor Half-Bridge. Fuente: Topologies for switched mode power supplies. STMicroelectronics .....	34
Ilustración 20 - Esquema básico convertidor Full Bridge. Fuente: Topologies for switched mode power supplies. STMicroelectronics .....	34
Ilustración 21 - Esquema básico convertidor Full Bridge con tanque de resonancia. Fuente: Phase-Shifted Full Bridge DC/DC Power Converter Design Guide. Texas Instruments.....	35
Ilustración 22 - Fase 1 Operación convertidor Full Bridge. Fuente: ZVS Phase Shift Full Bridge CFD2 Optimized Design. Infineon.....	36
Ilustración 23 - Fase 2 Operación convertidor Full Bridge. Fuente: ZVS Phase Shift Full Bridge CFD2 Optimized Design. Infineon .....	36
Ilustración 24 - Señales de control convertidor Full Bridge. Fuente: Phase-Shifted Full Bridge DC/DC Power Converter Design Guide. Texas Instruments.....	37
Ilustración 25 - Estructura de control Full Bridge y rectificación Síncrona. Fuente: Phase-Shifted Full Bridge DC/DC Power Converter Design Guide. Texas Instruments.....	38
Ilustración 26 - Diodos VS MOSFET en la rectificación del secundario. Fuente: Wikipedia.....	39

Ilustración 27 - Esquema de operación de la fuente de alimentación desarrollada en este trabajo .....	40
Ilustración 28 - Modelo 3D Panel frontal. Vista exterior .....	42
Ilustración 29 - Modelo 3D Panel frontal. Vista interior .....	42
Ilustración 30 - Modelo 3D Etapa de salida .....	43
Ilustración 31 - Terminal de 4 mm "banana" de 30A. Fuente: Aliexpress .....	43
Ilustración 32 - Modelo 3D Placa de botones y encoders.....	43
Ilustración 33 - Esquema eléctrico antirrebote de encoders.....	44
Ilustración 34 - Cadena de pulsos generada por los emcoders en el contador CD4013.....	44
Ilustración 35 - Nivel de tensión generado por el contador CD4013 al cambiar de sentido de giro. ....	45
Ilustración 36- Circuito implementado para amortiguar las señales de los botones de los encoders.....	45
Ilustración 37 - Pulsaciones del botón integrado en el encoder.....	45
Ilustración 38 - Implementación antirrebote botones BFUNC y BOUT.....	46
Ilustración 39 - Pulsaciones generadas por los botones BFUNC y BOUT, nótese que generan un nivel de tensión bajo .....	46
Ilustración 40 - PCB completa de los botones y encoders. Vista superior.....	47
Ilustración 41 - PCB completa de botones y encoders. Vista inferior .....	47
Ilustración 42 - Pantalla Nextion 3.2". Vista superior e inferior .....	48
Ilustración 43 - Interfaz diseñada en Nextion Editor .....	48
Ilustración 44- Interruptor AC 10A 21x15 mm. Fuente: Aliexpress .....	48
Ilustración 45 - Vista en Cura 3D del panel frontal en la cama de BQ Hephestos 2 .....	49
Ilustración 46 - Montaje final del panel frontal con la placa de botones y encoders, la etapa de salida y la pantalla. Vista interior .....	49
Ilustración 47. Montaje final panel frontal con cubrebotones, interruptor AC, terminales 4 mm y puerto USB. Vista exterior.....	50
Ilustración 48 - Modelo 3D Panel trasero con fuentes de alimentación. Vista interior.....	51
Ilustración 49 - Panel trasero. Vista exterior.....	51
Ilustración 50 - Conector AC 10 <sup>a</sup> . Fuente: Aliexpress .....	52
Ilustración 51 - Vista en Cura 3D del panel trasero en la cama de BQ Hephestos 2.....	52
Ilustración 52 - Montaje panel trasero con conector AC. Vista exterior .....	53
Ilustración 53 - Montaje panel trasero con fuentes de alimentación. Vista interior.....	53
Ilustración 54 - Esquema basico de alimentación de los circuitos auxiliares.....	54
Ilustración 55 - Fuentes de alimentación utilizadas. Izquierda: fuente de 12V 1A. Derecha: fuente de 12V 2A. Fuente: Aliexpress.....	54
Ilustración 56 - Convertidores aislados como alternativa para el controlador PFC. Izquierda: MORSUN. Derecha: MEAN WELL .....	55
Ilustración 57 - Modelo 3D caja exterior de acero de la fuente de alimentación.....	55
Ilustración 58 – Modelo 3D Lámina de acero inferior.....	56
Ilustración 59 - Modelo 3D Lámina de acero superior.....	56
Ilustración 60 - Modelo 3D montaje Filtro EMI y convertidor Boost PFC .....	57
Ilustración 61 - Modelo 3D etapa de entrada y filtro EMI .....	57
Ilustración 62 - Fusible 10A 38 mm HRC. Fuente: Aliexpress.....	58
Ilustración 63 - Varistor ZOV 20D431k 275VAC. Fuente: Aliexpress.....	58
Ilustración 64 - Esquema típico filtro EMI .....	59
Ilustración 65 - Esquema del arranque suave implementado .....	60
Ilustración 66 - Relé SONGLE SLA-12VDC-SL-A. Fuente: Aliexpress.....	60

Ilustración 67 - Terminal de crimpar Faston 6.3 mm. Fuente Aliexpress .....	60
Ilustración 68 - Pin Header 2x5. Fuente: LCSC .....	61
Ilustración 69 - PCB etapa de entrada y filtro EMI. Vista superior.....	61
Ilustración 70 - PCB etapa de entrada. vista inferior. Se observan las pistas estañadas .....	61
Ilustración 71 - Modelo 3D convertidor Boost PFC.....	62
Ilustración 72 - Rectificador de onda completa. Fuente: electronics-tutorials.ws .....	63
Ilustración 73 - Vista de la onda rectificada sin filtrar. Fuente: elprocus.com.....	63
Ilustración 74 - Vista de la onda rectificada con filtro C. Fuente: DiodeLab.com .....	63
Ilustración 75 - Puente de diodos KBU1510. Fuente: Aliexpress .....	64
Ilustración 76 - Esquema básico convertidor Boost PFC. Fuente: PFC Boost Converter Design Guide. Infineon.....	65
Ilustración 77 - Bobina Aiyima 2 mH. Fuente: Aliexpress .....	66
Ilustración 78 - Transistor MOSFET IRFP460. Fuente: Aliexpress .....	66
Ilustración 79 - Diodo doble MUR3060PT. Fuente: Aliexpress. ....	66
Ilustración 80 - Condensadores EPCOS 470uF 450V. Fuente: Aliexpress .....	67
Ilustración 81 - Espacio mínimo entre pistas para tensiones de 500V. PCB Saturn Design.....	67
Ilustración 82 - Corriente soportada por una pista de 6 mm de ancho y 35 um de espesor. PCB Saturn Design .....	68
Ilustración 83 - PCB completa convertidor Boost PFC .....	69
Ilustración 84 - PCB completa convertidor Boost PFC. Vista inferior. Nótese que se han cubierto las pistas más importantes con estaño.....	69
Ilustración 85 - Modelo 3D controlador PFC.....	70
Ilustración 86 - Controlador ICE3PCS01G. Fuente: Aliexpress .....	70
Ilustración 87 - Circuitería de selección de frecuencia. Fuente: Datasheet ICE3PCS01G .....	70
Ilustración 88 - Frecuencia según valor óhmico utilizado. Fuente: Datasheet ICE3PCS01G.....	71
Ilustración 89 - Transistor encargado de manejar el relé de arranque suave .....	71
Ilustración 90 - Circuitería del lazo de tensión. Fuente: Datasheet ICE3PCS01G.....	72
Ilustración 91 - Divisores de tensión para VSENSE y OVP .....	72
Ilustración 92 -Circuitería del lazo de corriente. Fuente: Datasheet ICE3PCS01G.....	73
Ilustración 93 - Driver IR2125. Fuente: Aliexpress .....	73
Ilustración 94 - Conexión típica driver IR2125. Fuente: Datasheet IR2125 .....	74
Ilustración 95 - PCB controlador PFC .....	74
Ilustración 96 - Montaje completo filtro EMI y convertidor PFC .....	74
Ilustración 97 - Modelo 3D Convertidor DC - DC ZVS.....	75
Ilustración 98 - Esquema implementado del primario del convertidor DC - DC.....	75
Ilustración 99 - Transistor MOSFET IRFP460. Fuente: Aliexpress .....	76
Ilustración 100 - Diodo MUR860. Fuente: Aliexpress .....	76
Ilustración 101 - Transformador de corriente PULSE PE-63587NL Fuente: PULSE Electronics ..	77
Ilustración 102 - Circuito de medida de corriente en el primario implementado .....	78
Ilustración 103 - Esquema del transformador .....	78
Ilustración 104 - Izquierda: transformador con el bobinado primario. Derecha: transformador completado .....	79
Ilustración 105 - Izquierda (Núcleo ETD49) Derecha (Núcleo PQ4040).....	79
Ilustración 106 - Terminales Faston hembra (Izquierda) y macho (derecha). Fuente: Aliexpress .....	80
Ilustración 107 - Bobina 33uH y 10A. Fuente: Aliexpress .....	80
Ilustración 108 - Pendiente deseada en la corriente del convertidor. Fuente: STEVAL-ISA172V2: 2 kW fully digital AC - DC power supply. ST Microelectronics .....	80

Ilustración 109 - Snubber RCD implementado.....	81
Ilustración 110 - Shunt 0.02 ohmios utilizado. Fuente: Aliexpress.....	81
Ilustración 111 - Circuito de medida de corriente en el secundario.....	82
Ilustración 112 - Secundario del convertidor DC - DC.....	82
Ilustración 113 - Transistor MOSFET IRFP90N20D.....	83
<i>Ilustración 114 - Esquema básico rectificación síncrona autónoma. Fuente: The Implication of Synchronous Rectifiers to the Design of Isolated, Single-Ended Forward Converters. Texas Instruments.</i> .....	83
Ilustración 115 - Condensadores Nichicon 470 uF 100V. Fuente: Aliexpress.....	84
Ilustración 116 - Capacidad de corriente para pistas de 10 mm y 35 um de espesor. PCB Saturn Design.....	84
Ilustración 117 - PCB Completo convertidor DC - DC. Vista superior.....	85
Ilustración 118 - PCB completo convertidor DC - DC. Vista inferior.....	85
Ilustración 119 - Conexión típica driver IR2110 en el lado de alta tensión. Fuente: Datasheet IR2110.....	86
Ilustración 120 - Modelo 3D del driver puente completo con aislamiento para el lado de alta tensión.....	87
Ilustración 121 - Driver MCP1404. Fuente: Microchip.....	87
Ilustración 122 - Transformador de pulsos P0584NL. Esquema (Izquierda) Aspecto físico (Derecha). Fuente: Datasheet P0584NL.....	87
Ilustración 123 - PCB Completa driver puente completo P0584NL.....	88
Ilustración 124 - Funcionamiento del driver de puente completo con dos señales simétricas al 50%.....	88
Ilustración 125 - Modelo 3D driver rectificación síncrona.....	89
Ilustración 126 - PCB completo del driver de rectificación síncrona.....	89
Ilustración 127 - Funcionamiento driver de rectificación síncrona con dos señales simétricas al 50%.....	89
Ilustración 128 - Modelo 3D controlador de temperatura.....	90
Ilustración 129 - Disipador de aluminio 40x40x80 mm. Fuente: Aliexpress.....	90
Ilustración 130 - Disipador CR101-50. Fuente: Mouser.....	91
Ilustración 131 - NTC 10K B3950. Fuente: Sparkfun.....	91
Ilustración 132 - Vista de montaje del termistor NTC.....	92
Ilustración 133 - Transistor MOSFET IRFZ44. Fuente: Aliexpress.....	92
Ilustración 134 - Características generales IRFZ44.....	92
Ilustración 135 - Ventilador Delta FFB0412SVHN. Fuente: Aliexpress.....	93
Ilustración 136 - Señales PWM en varios ciclos de trabajo.....	93
Ilustración 137 - Microcontrolador ATtiny 84. Fuente: Microchip.....	94
Ilustración 138 - Esquema de funciones por patillaje ATtiny84. Fuente: <a href="http://homemadehardware.com">homemadehardware.com</a> .....	94
Ilustración 139 - Diagrama de flujo del programa para el controlador de temperatura.....	95
Ilustración 140 - PCB completa del controlador de temperatura.....	96
Ilustración 141 - Señales PWM al 0% con una temperatura menor de 35°C.....	96
Ilustración 142 - Señales PWM al 25% para una temperatura entre 35 y 40°C.....	97
Ilustración 143 - Ilustración 136 - Señales PWM al 50% para una temperatura entre 40 y 45°C.....	97
Ilustración 144 - Ilustración 136 - Señales PWM al 75% para una temperatura entre 45 y 55°C.....	97

Ilustración 145 - Ilustración 136 - Señales PWM al 100% para una temperatura entre 55 y 65°C o más. ....	98
Ilustración 146 - Modelo 3D controlador del convertidor DC - DC.....	99
Ilustración 147 - Microcontrolador dsPIC33EV256GM002. Fuente: Microchip.....	99
Ilustración 148 - Regulador de tensión AMS1117-5.0 Fuente: Aliexpress.....	100
Ilustración 149 - Referencia de tensión MCP1501. Fuente: Microchip .....	100
Ilustración 150 - Se observa que la tensión AVDD. no pueden ser menor que VDD - 0.3V ni mayor que VDD + 0.3V. Fuente: Datasheet dsPIC33EV .....	100
Ilustración 151 - Referencia de tensión TL431A. Fuente: Aliexpress.....	101
Ilustración 152 - Esquema de implementación de TL431A.....	101
Ilustración 153 - Amplificador operacional MCP6V26. Fuente: Microchip.....	102
Ilustración 154 - Implementación de lectura de tensión en el secundario.....	102
Ilustración 155 - Generador de tensión negativa TC7660. Fuente: Microchip .....	103
Ilustración 156 - Esquema de implementación TC7660.....	103
Ilustración 157 - Amplificador de instrumentación INA126P. Fuente: Texas Instruments.....	104
Ilustración 158 - Esquema de implementación para la lectura de corriente en el secundario	104
Ilustración 159 - Diagrama interno ADP3605. Fuente: Analog Devices .....	105
Ilustración 160 - Circuito de activación del relé de salida.....	106
Ilustración 161 - Diagrama de flujo del programa para el controlador del convertidor DC - DC .....	106
Ilustración 162 - Configuración ADC MPLAB Code Configurator .....	108
Ilustración 163 - Salidas complementarias PWM 100KHz al 50% de ciclo de trabajo .....	108
Ilustración 164 - 120 nanosegundos aproximadamente de Dead Time .....	109
Ilustración 165 - Configuración puerto I2C Maestro MPLAB Code Configurator .....	109
Ilustración 166 - Configuración interrupción del timer 1 cada 100 ms MPLAB Code Configurator .....	110
Ilustración 167 - Esquema de control a implementar. Control por pico de corriente en el primario.....	111
Ilustración 168 - Diagrama de Bode del controlador con $F_s = 60000$ KHz.....	112
Ilustración 169 - Esquema de implementación SMPS Control Library. Fuente: Microchip .....	112
Ilustración 170 - PCB Completa del controlador DC - DC. Vista superior.....	113
Ilustración 171 - PCB Completa del controlador DC - DC. Vista inferior. ....	114
Ilustración 172 - Modelo 3D del controlador del panel frontal .....	114
Ilustración 173 - Microcontrolador ATmega328P. Fuente: Microchip .....	115
Ilustración 174 - Conversor UART - USB CH340G. Fuente: Aliexpress .....	115
Ilustración 175 - Esquema de implementación CH340G.....	116
Ilustración 176 - Conversor UART - USB FTDI FT232RL. Fuente: Aliexpress .....	116
Ilustración 177 - Conversor UART - USB MCP2200. Fuente: Microchip.....	116
Ilustración 178 - Regulador LM7805. Fuente: Aliexpress .....	117
Ilustración 179 - Diagrama de flujo del programa para la gestión del panel frontal .....	117
Ilustración 180 - Trama UART. Fuente: solitontech.com .....	118
Ilustración 181 - Tabla de preescala de reloj para el timer 1. Fuente: Datasheet ATmega328P .....	119
Ilustración 182 - Esquema de jerarquía protocolo I2C. Fuente: electronicshub.org .....	121
Ilustración 183 - Protocolo I2C. Fuente: Sparkfun .....	121
Ilustración 184 - PCB Completa control del panel frontal. Vista superior .....	122
Ilustración 185 - PCB Completa control del panel frontal. Vista inferior.....	122
Ilustración 186 - Interfaz completa del programa en LabView .....	123

Ilustración 187 - Gestión del puerto USB .....	123
Ilustración 188 - Cadenas escritas y recibidas para depuración .....	124
Ilustración 189 - Controles de consigna .....	124
Ilustración 190 - Gráficas de tensión y corriente .....	124
Ilustración 191 - Indicadores de tensión, corriente y potencia .....	125
Ilustración 192 - Módulo de carga de baterías .....	125
Ilustración 194 - Panel frontal del bloque de gestión USB.....	126
Ilustración 195 - Esquema básico filtro EMI.....	133
Ilustración 196 - Procedimiento de cálculo de filtros EMI .....	133
Ilustración 197 - Pérdidas MOSFET Conducción VS Conmutación.....	135
Ilustración 198 - Esquema de los parámetros de un transformador .....	138
Ilustración 199 - Explicación del efecto piel en un conductor .....	139
Ilustración 200 - Esquema de consideraciones para el cálculo de disipadores de calor .....	142
Ilustración 201 - Se puede considerar el convertidor como un reductor básico .....	146
Ilustración 202 - Controlador implementado en el programa .....	147
Ilustración 203 - Diagrama de Bode del controlador con una $F_s = 60000$ Hz, presenta estabilidad y adecuado margen de fase.....	148
Ilustración 204 - Logotipo FUSION 360 .....	155
Ilustración 205 - Logotipo GRABCAD.....	155
Ilustración 206 - Logotipo EAGLE .....	155
Ilustración 207 – Interfaz de sincronización con FUSION 360 en EAGLE .....	156
Ilustración 208 - Logotipo MATLAB.....	156
Ilustración 209 - Logotipo SATURN PCB Design .....	157
Ilustración 210 - Logotipo LabVIEW .....	157
Ilustración 211 - Interfaz Arduino IDE .....	157
Ilustración 212 - Placa de desarrollo Arduino Nano.....	158
Ilustración 213 - Logotipos MPLAB X IDE y MPLAB Code Configurator .....	158
Ilustración 214 - Programadora PICKit3 .....	158
Ilustración 215 - Interfaz Nextion Editor .....	159
Ilustración 216 - Logotipo Cura 3D.....	159
Ilustración 217 - Entorno de trabajo BitScope. Fuente: BitScope.com.....	165
Ilustración 218 - Dispositivo BitScope. Fuente: BitScope.....	165
Ilustración 219 - Generador de señales FY3200. Fuente: Aliexpress .....	165
Ilustración 220 - Multímetro AN800. Fuente: Aliexpress.....	166
Ilustración 221 - Osciloscopio Hantek 5102P 100 MHz .....	166
Ilustración 222 - Entorno de programación del controlador y comunicaciones USB .....	167
Ilustración 223 -Prototipo de la fuente completa.....	168
Ilustración 224 - Prototipo de la fuente completa.....	168
Ilustración 225 - Prototipo original del filtro EMI (Arriba), segunda versión (Abajo) .....	168
Ilustración 226 – Modificaciones realizados en convertidor PFC .....	169
Ilustración 227 - Daños en el MOSFET del convertidor PFC. Se observa grieta en el cuerpo del transistor .....	170
Ilustración 228 -MOSFET dañado del convertidor PFC fuera de la placa. Los daños recibidos provocan que se fragmente en dos mitades.....	170
Ilustración 229 - Modificaciones realizados en placa de botones y encoders.....	171

## Resumen

Este proyecto nace con el objetivo de estudiar el diseño y la fabricación de un prototipo de fuente de alimentación de alta potencia basada en tecnología de conmutación y controlada mediante microcontroladores. A lo largo del trabajo se analizarán las topologías más utilizadas en el diseño de este tipo de instrumentos, así como la viabilidad para su fabricación.

En primer lugar, se estudiará la implementación de un sistema de corrección del factor de potencia y el controlador comercial más adecuado para esta aplicación. Después de esta primera etapa, se desarrollará el convertidor DC/DC del tipo ZVS (Zero Voltage Switching) que más se ajuste a las prestaciones deseadas en términos de potencia y corriente en la salida. Esta última etapa, será controlada mediante un lazo de control implementado en un microcontrolador.

El proyecto también incluye el desarrollo de un control de temperatura capaz de modificar la velocidad de ventiladores basado en técnicas PWM, con el objeto de reducir el ruido y las pérdidas asociadas a la refrigeración.

Se desarrollará también, la electrónica que permita la comunicación de la fuente con un PC mediante enlace USB, así como un programa LabView que sea capaz de mandar y leer consignas de tensión y corriente de forma automática. Además, dicho programa facilitará establecer funcionalidades tales como la carga de baterías.

## Abstract

This project focuses in the design and development process of a high power, high efficiency, switched mode power supply. It will cover the most common topologies used in these systems and their control based on microcontrollers.

In the first place, this paper will study the implementation of a power factor correction device that will be able to reach unity power factor in the whole power range of the instrument.

After this first step, an implementation of a DC – DC converter of the soft switching kind will be covered, as they have proven to be the most efficient and reliable throughout the history of power supplies.

The main goal is to develop a complete laboratory grade power supply that could be programmed to adjust its output voltage and limit its output current. This is to be implemented with a control loop into a microcontroller.

In addition, the system must regulate the working temperature of its components via active cooling based on PWM controlled fans, that will reduce audible noise and power losses.

Likewise, the electronics needed to connect the system to a PC via USB will be developed. This piece of software will be developed in LabVIEW and it will be able to control the output voltage and current commands, as well as saving the readings of these variables into a file and automate the charging process of batteries.



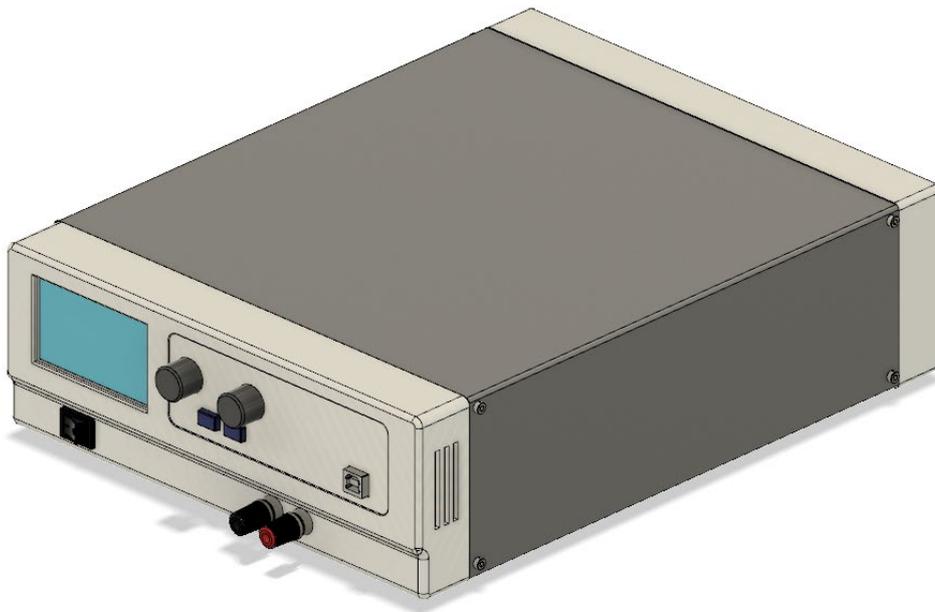
## 1. Introducción

En el día a día del desarrollo de aplicaciones electrónicas, disponer de una fuente de alimentación que permita variar la tensión y limitar la corriente en la salida, es fundamental.

La fuente de alimentación programable, es uno de los elementos básicos en cualquier laboratorio, ya sea para realizar pruebas con prototipos, analizar la carga de nuevas celdas de baterías, comprobar la eficiencia de convertidores DC – DC, etc.

Además, sus utilidades no solo se aplican a un entorno de laboratorio, si no que un control a través de interfaces modernas USB, RS - 232 o Ethernet, permite extender su uso en entornos de producción donde realizar pruebas o validar parámetros de forma remota y cíclica.

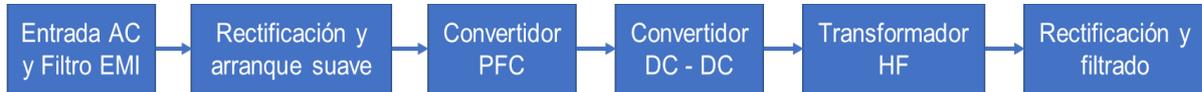
Es por ello, que este proyecto se ha centrado, no sólo en el desarrollo de la propia fuente de alimentación, si no de una interfaz que la dote de comunicaciones con el exterior a través del estándar USB.



*Ilustración 1 – Modelo 3D Fuente completa: Vista exterior*

## 2. Antecedentes

### 2.1. Estructura de una fuente de alimentación conmutada



*Ilustración 2 - Esquema fuente de alimentación conmutada*

Las fuentes de alimentación conmutadas son aparatos eléctricos de cierta complejidad a primera vista, pero que pueden dividirse en bloques muy diferenciados entre sí más fáciles de trabajar por separado, tal y como se representa en la Ilustración 2.

El primer bloque, generalmente, consisten en un filtro de interferencias electromagnéticas (EMI), colocado inmediatamente después de la entrada de corriente alterna para evitar verter armónicos a la red, derivados de los procesos de conmutación asociados al propio funcionamiento de la fuente.

Acto seguido suele aparecer la etapa del rectificador y filtro que convierte la corriente alterna en corriente continua. Generalmente, las fuentes de alimentación de baja potencia suelen incluir un arrancador suave (encargado de limitar el pico de corriente de conexión) implementado mediante un termistor NTC en serie para limitar la corriente de *inrush* y evitar que se activen las protecciones de la instalación eléctrica de suministro, o que se estropeen componentes internos de la propia fuente.

En modelos de mayor potencia, el termistor se suele reemplazar por una resistencia en paralelo con un relé. Durante el proceso de encendido, primeramente, la resistencia queda en serie con el circuito; en la que una vez ha pasado un cierto periodo de tiempo (el que se estima suficiente para limitar los picos de corriente de conexión), se cierra el relé y deja cortocircuitada la resistencia.

Una vez que se ha rectificado la corriente, se puede conectar directamente a los bloques posteriores, aunque el factor de potencia de estas fuentes es muy pobre. En modelos de bajas potencias, esto es lo habitual, generalmente están construidas pensando en el coste y tampoco representan una amenaza para la red eléctrica.

A partir de 200W de potencia nominal las fuentes empiezan a incluir tecnologías de corrección del factor de potencia, tanto pasivas como activas; estando las primeras, a día de hoy, obsoletas.

Una vez que se ha asegurado un nivel de continua estable, se pasa a un convertidor DC – DC; existiendo muchas variantes según la potencia y la eficiencia que se desee obtener. Se basan, generalmente, en transistores MOSFET, que deben manejar un transformador de ferrita a altas frecuencias (HF) mediante técnicas PWM.

Posteriormente, en el lado secundario del transformador se produce un nuevo rectificado y filtrado al nivel de tensión deseado mediante un lazo de control encargado de regular dicha tensión y la intensidad máxima que entrega la fuente.

A continuación, se pasa a ampliar la información sobre cada una de las etapas anteriormente mencionadas

## 2.2. Filtrado EMI

Las fuentes de alimentación conmutadas son, por norma general, la mayor fuente de ruido de cualquier sistema en el que se integren. Especialmente las perturbaciones relacionadas con grandes cambios de tensión y corriente en la conmutación.

Los filtros EMI, tienen por objeto reducir el “ruido por conducción” que se asocia al funcionamiento de algunos dispositivos. Dentro del ruido por conducción se diferencia entre ruido diferencial y ruido en modo común.

La fuente del ruido en modo diferencial es la corriente conmutada por los MOSFET o diodos. Grandes cambios de tensión y la corriente que los condensadores vierten a tierra son los causantes del ruido en modo común.

### 2.2.1. Ruido en modo diferencial

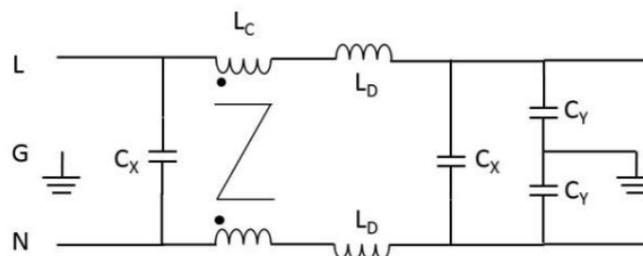
Es el que se mide entre la fase de alimentación y el neutro. Se produce por el acoplamiento magnético y trata de encontrar un camino entre estas dos líneas. Puede provocar problemas en condensadores y transformadores si su amplitud es demasiado alta.

### 2.2.2. Ruido en modo común

Se mide entre fase y tierra. La mayor fuente de ruido en modo común son las corrientes parásitas de condensadores conectados a tierra. Los tres contribuyentes a este fenómeno son la capacitancia entre los transistores y el disipador, capacidades entre los devanados de transformadores y la capacitancia entre cables.

Consiste en ruido en forma de pulsos de alta frecuencia que puede venir del transformador de alta frecuencia y los condensadores de filtro. También puede proceder por la capacitancia del drenador en los MOSFET, al conmutar, se producen caídas de tensión que generan este fenómeno.

### 2.2.3. Filtro más extendido



*Ilustración 3 - Esquema típico filtro EMI. Fuente: Practical Design of a Passive EMI Filter for Reduction of EMI Generation*

Hay varios tipos de filtros, los hay que incluyen filtrado en modo diferencial o no. También se pueden realizar varios diseños en modo común, generalmente de segundo y tercer orden, estando más extendido el de segundo orden por su sencillez y coste más contenido.

En la Ilustración 3 se observa el filtro que se ha desarrollado para este trabajo. Consiste en una bobina ( $L_C$ ) y dos condensadores en modo común ( $C_Y$ ). Además, se han añadido dos bobinas ( $L_D$ ) y dos condensadores ( $C_X$ ) en modo diferencial.

### 2.3. El factor de potencia

El factor de potencia se define simplemente como la relación entre la potencia real, y la potencia aparente:

$$PF = \frac{\text{Potencia real (W)}}{\text{Potencia aparente (VA)}}$$

Donde la potencia real es el producto de la media de corriente y tensión instantáneos y la potencia aparente es el producto de la corriente y tensión RMS. Si la corriente y la tensión tienen forma sinusoidal y están en fase, el factor de potencia es 1. Si ambas formas de onda son sinusoidales, pero no están en fase, el factor de potencia es el coseno del ángulo de desfase entre ellas.

Generalmente, se suele estudiar el factor de potencia con cargas como resistencias, bobinas o condensadores, con formas de onda sinusoidales, y una respuesta lineal.

Las fuentes de alimentación conmutadas presentan una impedancia, no lineal dada la presencia de un rectificador y un condensador de almacenamiento, capaz de mantener la energía hasta el pico de tensión sinusoidal.

Al llegar a este pico, se produce una caída de la tensión y se demanda corriente de la red durante un breve intervalo de tiempo. Generalmente entre un 10 y un 20% del ciclo de la señal, lo que quiere decir que este pico de corriente es entre 5 y 10 veces mayor a la corriente media demandada.

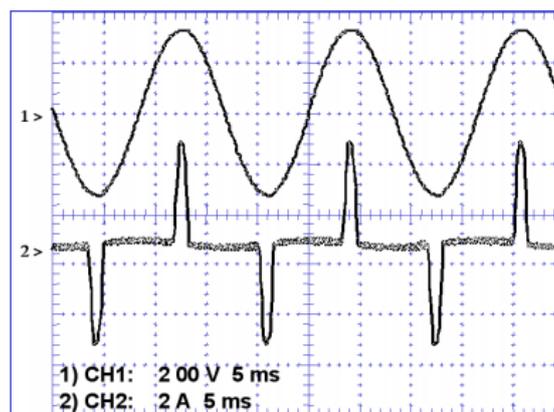


Ilustración 4 - Arriba: Tensión. Abajo: Corriente. Se observa que la tensión solo cae en el centro de la señal. Fuente: [powerelectronicstips.com](http://powerelectronicstips.com)

Al observar la forma de onda representada en la ilustración 4, se deduce que la corriente y la tensión, aunque tremendamente distorsionadas, están en fase. Esto podría llevar a la conclusión de que el factor de potencia de la carga es 1, resultado que es rotundamente falso.

Para realizar un cálculo correcto del factor de potencia debemos fijarnos, no solo en el coseno de  $\varphi$  de la carga, sino también en la distorsión por el contenido armónico que esta produce ( $\cos \theta$ ).

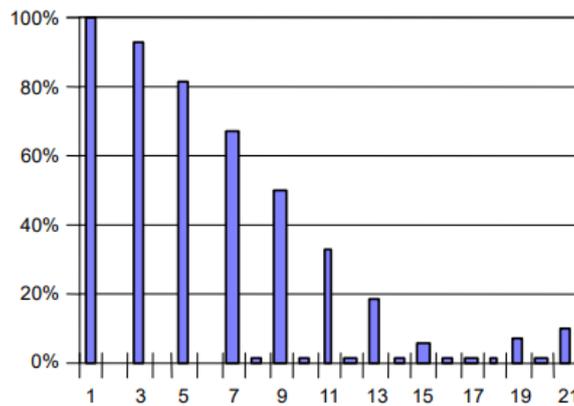


Ilustración 5 - Carga armónica en la línea sin corrección f.d.p. Fuente: powerelectronicstips.com

De esta manera, se deduce que el cálculo de la potencia real viene determinado por:

$$P_{in} = V_{in} (RMS) \times I_{in} (RMS) \times \cos \varphi \times \cos \theta$$

De esta manera se puede estimar que el factor de potencia de la señal anterior es aproximadamente 0,6.

Con los cálculos anteriores, se puede confirmar que la distorsión producida por cargas no lineales en la red, como es el caso de las fuentes de alimentación conmutadas, tiene grandes implicaciones en el factor de potencia.

Para medir esta distorsión, se utiliza la fórmula de la distorsión armónica total (THD) con respecto a una sinusoidal fundamental, que luego se utilizará para calcular su valor absoluto ( $\cos \theta$ ), que demuestra que la distorsión debe ser 0 para que el factor de potencia sea unidad.

$$THD (\%) = 100 \times \sqrt{\sum_{p=2}^{\infty} \left(\frac{I_p^2}{I^2}\right)} \quad \cos \theta = \sqrt{\frac{1}{1 + THD^2}}$$

En la ilustración 6 se aprecia una forma de onda de corriente sin distorsión dado contenido armónico nulo (a parte del fundamental), obteniéndose un factor de potencia unidad.

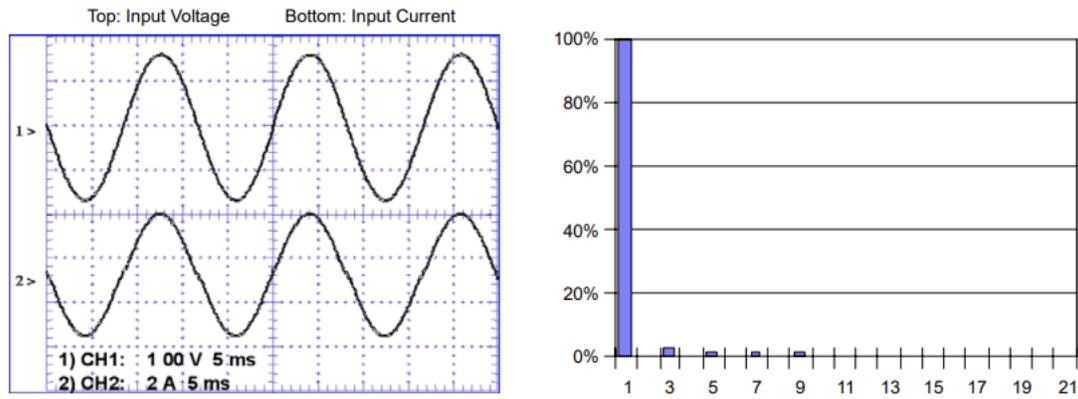


Ilustración 6 - Izquierda: Forma de onda sin distorsión. Derecha: El nivel de armónicos es el de la frecuencia fundamental

Existen varios métodos de corrección del factor de potencia, como se explicará más adelante, que responden a la evolución de la tecnología y reglamentos. Desde la corrección pasiva (también conocida como filtrado pasivo) mediante bobinas y condensadores, hasta convertidores activos (también conocidos como filtros activos) que modelan la forma de onda de la corriente en tiempo real según la carga demandada.

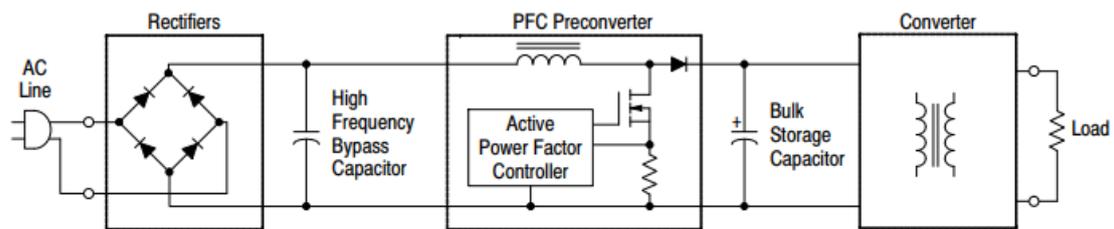


Ilustración 7 - Esquema típico de un convertidor Boost PFC. Fuente: [powerelectronicstips.com](http://powerelectronicstips.com)

En la ilustración 8 se compara la forma de onda obtenida según la tecnología de corrección utilizada:

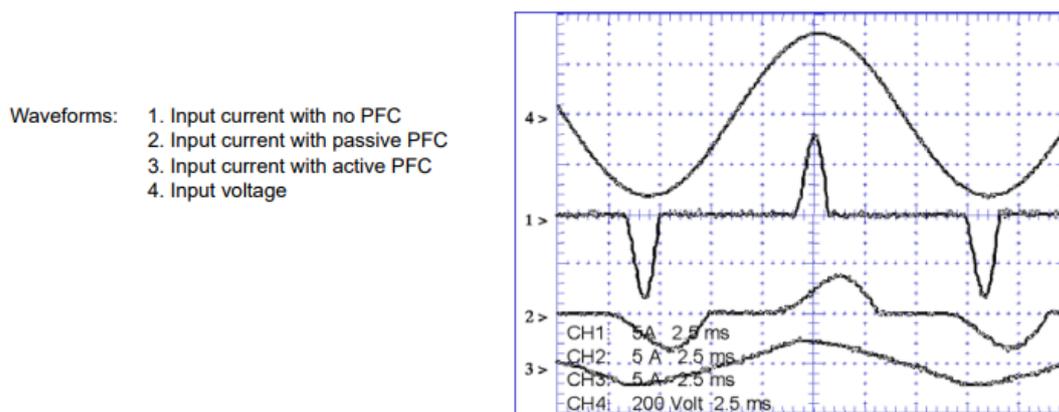


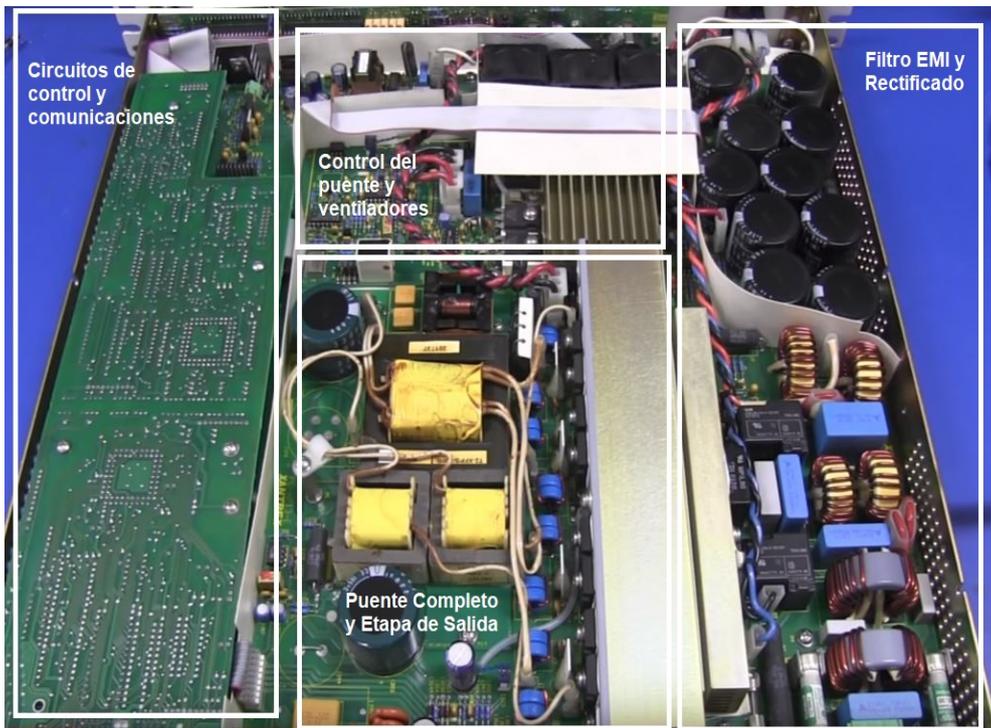
Ilustración 8 - Formas de onda de corriente con diferentes métodos de corrección. Fuente: [powerelectronicstips.com](http://powerelectronicstips.com)

### 2.3.1. Métodos de corrección f.d.p

En el pasado, las fuentes de alimentación conmutadas no incorporaban ningún tipo de corrección del factor de potencia, dando como resultado consumos elevados de la red.

Revisando los diseños de fuentes de la época de los años 90, se puede observar que se incluían filtros de alta frecuencia para evitar verter los armónicos de alta frecuencia generados por la conmutación de estos aparatos.

Sin embargo, se utilizaba un rectificado tradicional de la corriente alterna que entraba a la fuente, del que se obtenían unos 300VDC sin regulación e inmediatamente después, se colocaba el convertidor DC-DC encargado de proporcionar aislamiento galvánico y regular la salida en corriente continua.



*Ilustración 9 - Estructura fuente de alimentación Xantrex 300V4A. Fuente: EEVblog*

### 2.3.2. Métodos de corrección f.d.p – Corrección Pasiva

Actualmente existen diversos circuitos para la corrección pasiva del factor de potencia utilizando combinaciones de condensadores y bobinas. Dado que este método no es el objeto de estudio de este trabajo final de grado, sólo se van a comentar los dos métodos más extendidos a lo largo del tiempo en la implementación de las fuentes de alimentación conmutadas.

Los circuitos consisten en colocar una bobina en el lado de corriente continua o alterna, justo después o antes del rectificador; como muestran respectivamente las ilustraciones 10 y 11

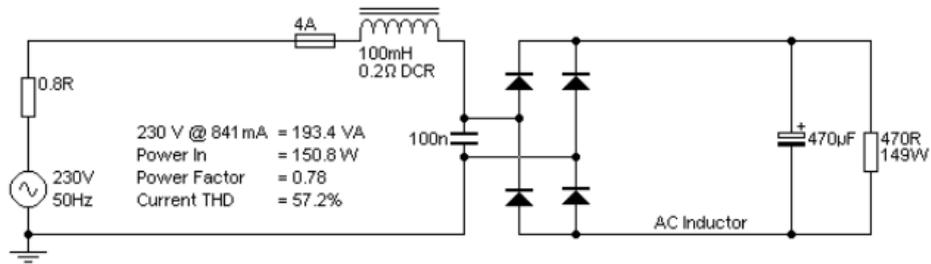


Ilustración 10 - Corrección pasiva f.d.p. con bobina en el lado de alterna. Fuente: sound-au.com

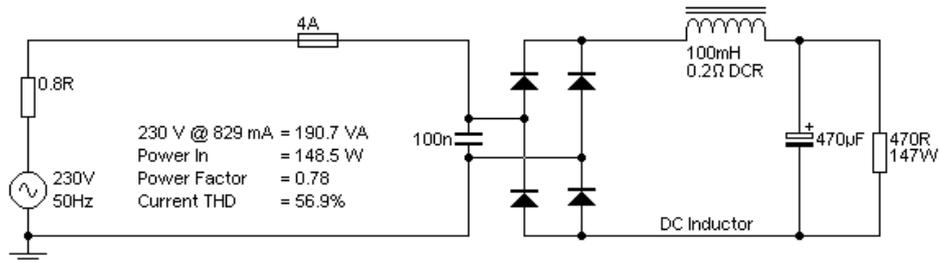


Ilustración 11 - Corrección pasiva f.d.p. con bobina en el lado de continua. Fuente: sound-au.com

En la ilustración 12, se observa que, si se coloca la bobina en el lado de alterna, justo antes del rectificador, se producen oscilaciones en el cambio de polaridad de la bobina al conectarse un condensador de filtro EMI. Si se conectase solo la bobina, sin condensador, oscilaría a su frecuencia de resonancia, suficientemente grande como para crear problemas de interferencias EMI. Al conectar un condensador, se mantienen estas oscilaciones a 1,59KHz según la imagen.

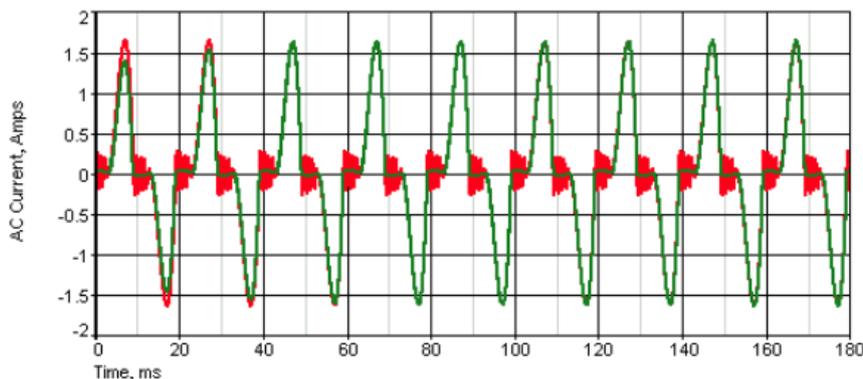
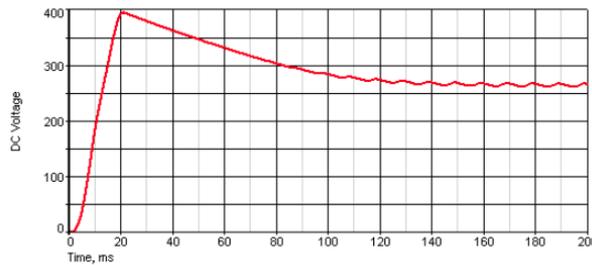


Ilustración 12 - Comparación corriente en la bobina en lado AC (Rojo) y en lado DC (Verde). Fuente: sound-au.com

Conectar la bobina en el lado de alterna, tiene la ventaja de que no se produce saturación en el núcleo de la misma al no existir un nivel de continua, lo que permite que la bobina sea más pequeña. Por el contrario, nos encontramos el problema de oscilaciones anterior.

Al colocarla en el lado de continua, se eliminan estas oscilaciones, pero al producirse saturación, la bobina debe ser sustancialmente mayor. Por otro lado, la respuesta característica en el encendido puede ser muy pronunciada. Este es un fenómeno relacionado con una oscilación de baja frecuencia creada por la bobina junto al

condensador de filtro. En la imagen se puede apreciar el pico de tensión en el encendido, el decaimiento y la oscilación producida de unos 20Hz.



*Ilustración 13 - Pico de tensión en el encendido*

El pico de tensión es independiente de la carga que se conecta, pero la velocidad del decaimiento aumenta con la corriente demandada. Por ejemplo, con una bobina de 10H y un condensador de 47 $\mu$ F, el pico se reduce a 240VDC, pero la tensión estable media obtenida es de 205 VDC, ya que el comportamiento de este tipo de filtro es aproximadamente:

$$VDC\ out = 0,9 \times VAC\ in$$

Cuando la bobina es lo bastante grande, la corriente es completamente “continua”, nunca baja a cero. Bajo esta condición, la corriente tiene forma de señal cuadrada. Las bobinas más pequeñas funcionan en un “modo discontinuo”, en el que la corriente baja a cero cada medio ciclo de alterna.

Para obtener buenos resultados, no caer en este modo discontinuo y reducir lo suficiente el pico de tensión, la bobina debe ser lo bastante grande (del orden de varios Henrios).

Al aumentar tanto la bobina, el condensador se vuelve irrelevante, por ejemplo, pasar de 4.7 $\mu$ F en el caso anterior, a 47nF solo incrementaría el rizado 3db.

En la práctica, ningún fabricante va a instalar una bobina de varias decenas de Henrios en un producto comercial, dado su elevado coste y tamaño.

La tensión media a la salida y el rizado obtenido, son idénticos independientemente de que la bobina se instale en el lado de alterna o el de continua. Por esta razón, el coste y para evitar el pico de tensión en el lado de continua, la bobina se coloca en el lado de alterna en la mayoría de los casos.

Esto ayuda, no solo a instalar bobinas más pequeñas, sino también a dimensionar los componentes que vienen después a una tensión menor al evitar el pico inicial.

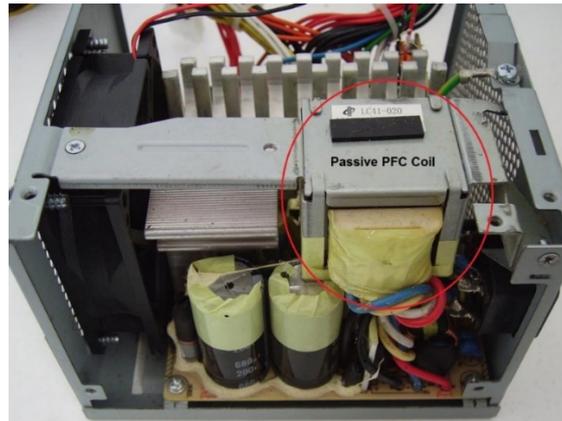


Ilustración 14 - Fuente de alimentación de PC con corrección pasiva del f.d.p. Fuente: sound-au.com

### 2.3.3. Métodos de corrección f.d.p – Convertidores modernos

Actualmente, no se utiliza una corrección del factor de potencia pasivo en fuentes de alimentación conmutadas debido a que el coste de un sistema de corrección activo es muy similar y ofrece mucho más rendimiento, a la vez que reduce el peso del dispositivo considerablemente.

Las topologías más populares son el convertidor DC – DC elevador de tensión (Boost) y un derivado del anterior bifásico, llamado intercalado (interleaved) en el que se acoplan dos convertidores elevadores en paralelo, desfasados 180° entre sí.

#### Convertidor elevador de tensión (Boost)

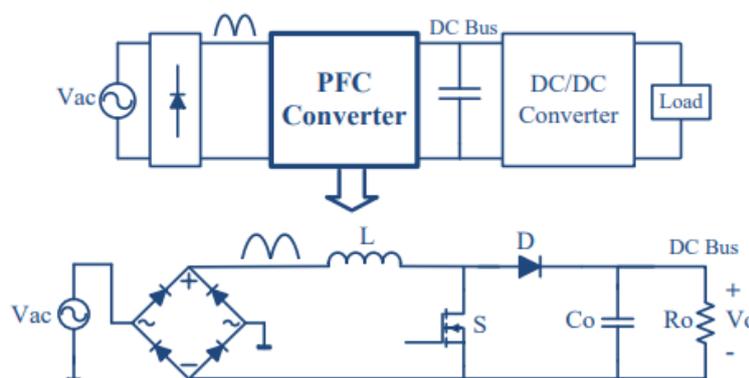


Ilustración 15- Esquema básico Boost PFC. Fuente: Infineon

El convertidor elevador de tensión, o *Boost Converter*, es la topología que más se ha popularizado como solución para corregir el factor de potencia en fuentes de alimentación conmutadas.

Dicho convertidor funciona cargando una bobina, simplemente, haciendo circular por ella la corriente demandada por el circuito. Posteriormente, se acciona un transistor a

modo de interruptor, que conecta la bobina a tierra, descargándola y generando un pico de tensión al oponerse la bobina a este cambio, mucho mayor que la tensión de entrada.

Dicho pico de tensión es rectificado a través de un diodo y un condensador de filtro a la salida. La relación entre la tensión de salida y la de entrada, es inversamente proporcional al ciclo de trabajo del transistor:

$$\frac{V_{out}}{V_{in}} = \frac{1}{1 - DT}$$

Una vez explicado su funcionamiento, podemos entender porque se ha hecho tan popular. La forma de onda de tensión que suministra la red puede variar enormemente en función de la distancia a la que nos encontremos de un centro de distribución, o puede sufrir picos de tensión repentinos en caso de avería o fallos en su funcionamiento.

Por ello, los fabricantes decidieron establecer un nivel de tensión al que, generalmente, funcionan todas las fuentes con corrección activa del f.d.p, llamada *Tensión de bus o línea*. Esta tensión es 400 VDC aproximadamente.

Una vez que se ha fijado un nivel de tensión por encima de cualquiera que pueda venir de la red, se puede inmunizar el circuito elevador a la tensión de entrada. Es decir, además de ser una solución a las variaciones de la red, también permite usar el mismo circuito en países con 120 VAC y 230 VAC.

Esto es posible porque dicho circuito, funciona con un controlador en lazo cerrado que asegura siempre una consigna de tensión constante a la salida, ajustando el ciclo de trabajo del transistor y en función de la corriente que circula por la bobina.

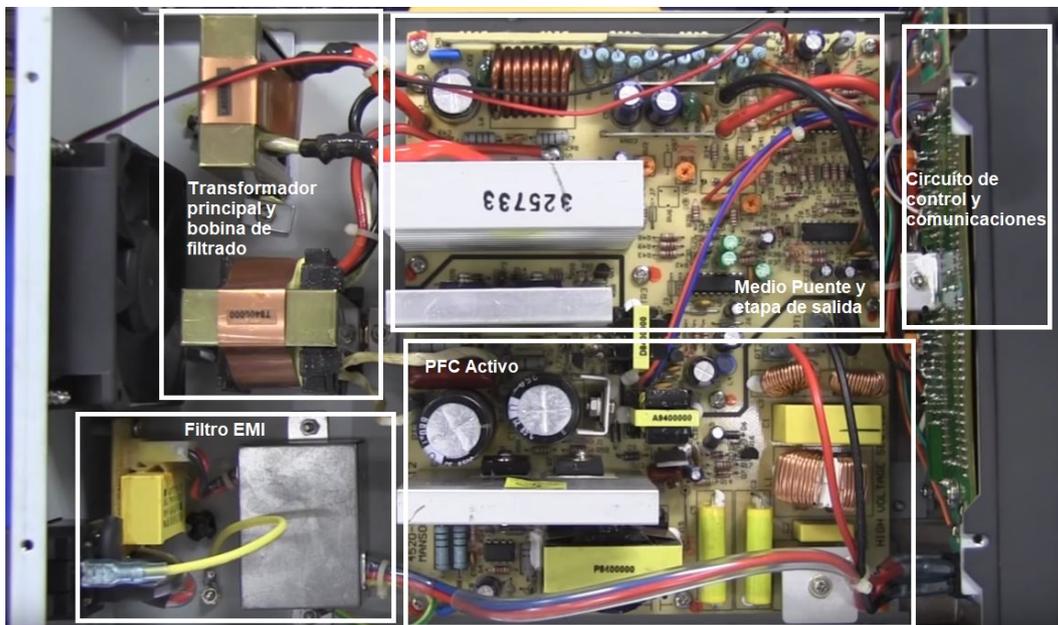


Ilustración 16 – Estructura fuente de alimentación Manson SPS-9400. Fuente: EEVblog

## Modos de operación

El convertidor elevador de tensión puede funcionar en tres modos distintos: modo de conducción continua (CCM), modo de conducción crítica (CrCM) y modo de conducción discontinua (DCM).

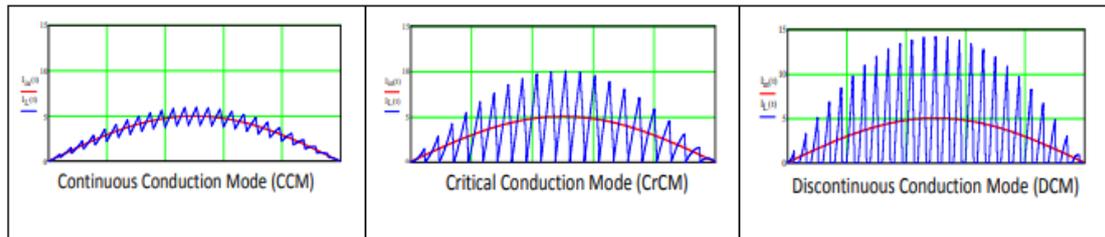


Ilustración 17 - Modos de operación convertidor Boost PFC. Fuente: PFC boost converter design guide. Infineon.

Comparando el modo DCM frente a los demás, es más sencillo que CrCM, ya que puede operar en un modo de frecuencia fija. Por el contrario, presenta la desventaja de tener el pico de corriente más alto frente a CrCM y CCM sin una ventaja de rendimiento que lo justifique. Por esa razón se van a estudiar los otros dos modos de conducción y se dejara a un lado la conducción DCM.

El modo de conducción CrCM es un caso especial de CCM, donde la operación se controla para estar entre CCM y DCM. Generalmente usa un modo de control que cambia con los eventos que se producen en la carga. De esta manera, va variando el tiempo de reset de la bobina y la frecuencia para mantener el convertidor en este modo. Básicamente, el controlador detecta el cruce por cero de la corriente de la bobina para comenzar el disparo del siguiente ciclo de conmutación.

El rizado de la corriente de la bobina (pico de corriente) en CrCM es dos veces el valor medio, lo que incrementa sustancialmente el valor RMS de la corriente que circula por el MOSFET y la corriente en el apagado de la conmutación. Pero como el ciclo comienza cuando la corriente cruza por cero, la corriente de encendido del MOSFET se elimina, así como la corriente de recuperación inversa del diodo.

El hecho de que CrCM presente un rizado tan grande, hace que sea recomendable evitarlo para diseños de gran potencia y su impacto en el filtro EMI requerido.

Por lo demás, las ecuaciones que se utilizan para diseñar un modo de operación CrCM son idénticas a las del modo CCM, diferenciándose en el rizado de la corriente, que afecta a las pérdidas globales de conmutación.

El modo de operación CCM requiere de una bobina mayor, ya que debe estar diseñada para un rizado de la corriente del 20-40% de la corriente media. Lo que presenta una serie de ventajas:

- La corriente de pico es menor, la corriente RMS con una forma de onda trapezoidal se reduce respecto a una onda triangular. Esto resulta en menores pérdidas de conducción.
- Las pérdidas de apagado en la conmutación son menores debido al menor valor de la corriente.

- El rizado de alta frecuencia puede ser reducido de manera más eficaz por el filtro EMI a una menor amplitud.

Por otro lado, se requieren dispositivos mejores, ya que el MOSFET debe resistir pérdidas de encendido mayores, que se ven acentuadas por las pérdidas de recuperación inversa del diodo dada la gran carga  $Q_{rr}$ .

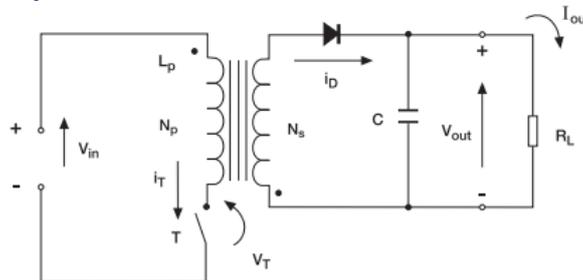
En conclusión, se podría decir que, para una aplicación de baja potencia, el modo de conducción CrCM tiene la ventaja de unas pérdidas menores, pero a partir de ciertas potencias el rizado puede ser un problema para el filtro EMI de la fuente. Por lo que, para un diseño de alta potencia, el modo de operación CCM es el adecuado y es el que se ha diseñado en este trabajo.

## 2.4. Convertidor DC - DC

El convertidor DC – DC de una fuente de alimentación conmutada, se encarga de manejar un transformador para proporcionar aislamiento galvánico de la red mediante el control de señales PWM. Después del transformador se colocan diodos y condensadores que filtran la salida.

Se explican a continuación, algunas de las topologías más utilizadas según la potencia requerida y la eficiencia que se quiera conseguir.

### 2.4.1. Convertidor Flyback



*Ilustración 18 - Esquema básico convertidor Flyback. Fuente: Topologies for switched mode power supplies. STMicroelectronics*

El convertidor Flyback, es una topología muy utilizada dado su bajo coste y sencillez. Se utiliza en fuentes que van desde los 30W hasta los 250W. Es el tipo de convertidor más simple, en el que solo hay un transistor encargado de manejar el transformador. Su control también resulta muy accesible.

### 2.4.2. Convertidor Medio puente (Half Bridge)

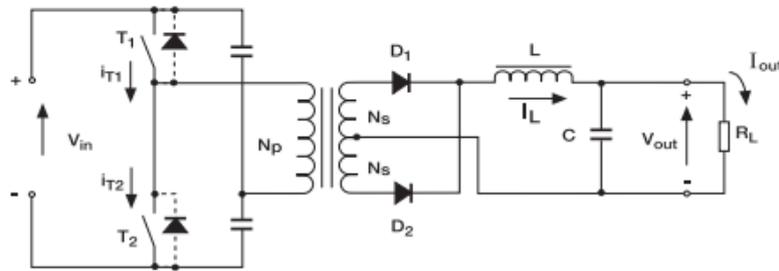


Ilustración 19 - Esquema básico convertidor Half-Bridge. Fuente: *Topologies for switched mode power supplies. STMicroelectronics*

Este circuito consiste en dos transistores que forman una onda cuadrada bipolar en el transformador. Se utilizan dos condensadores en serie para fijar un punto medio de tensión sobre la que se produzca la conmutación.

Puede desarrollar potencias de 500 – 1000W y se usa en fuentes de cierta calidad. Tiene la desventaja de requerir un driver bipolar en el primario para su operación, y según el driver utilizado, podrían aparecer desequilibrios en el tiempo de conmutación de los transistores.

### 2.4.3. Convertidor Puente Completo (Full Bridge)

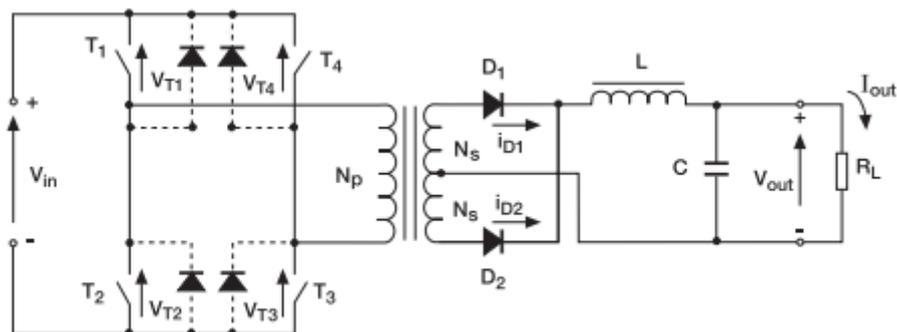


Ilustración 20 - Esquema básico convertidor Full Bridge. Fuente: *Topologies for switched mode power supplies. STMicroelectronics*

El convertidor de puente completo o *Full Bridge*, es una configuración de alta potencia, que se usa en aplicaciones que requieran más de 1KW, ya que tiene comportamiento térmico mejor que los demás circuitos, al incorporar más transistores.

Aunque su operación es algo más compleja, permite utilizar técnicas de control más avanzadas, que afectan positivamente a la eficiencia del convertidor, como se explicará posteriormente, ya que se ha decidido implementar este convertidor.

### 2.4.4. ZVS (Zero Voltage Switching)

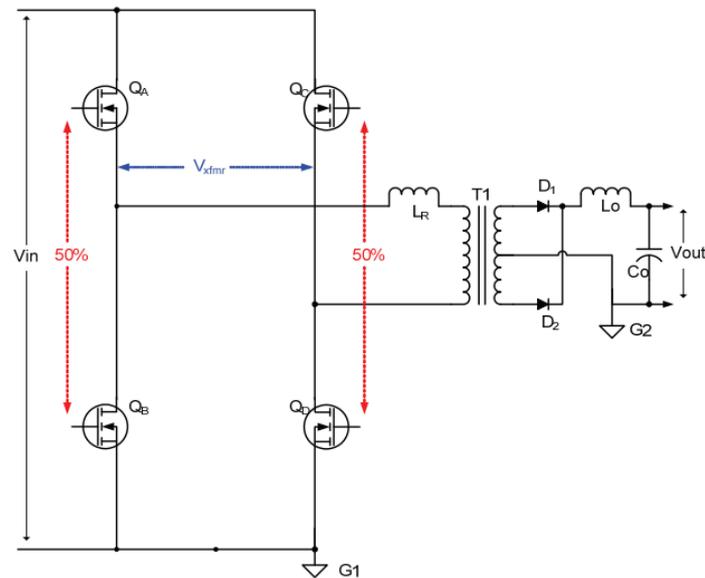


Ilustración 21 - Esquema básico convertidor Full Bridge con tanque de resonancia. Fuente: *Phase-Shifted Full Bridge DC/DC Power Converter Design Guide. Texas Instruments.*

Esta topología es la más extendida cuando se trata de implementar un convertidor de puente completo, ya que como veremos, reduce considerablemente las pérdidas y picos de tensión asociados a las conmutaciones.

Se trata de los mismos cuatro interruptores que antes, ya sean MOSFET o IGBTs los semiconductores de potencia que forman un puente completo en el primario con la adición de una bobina  $L_r$  en serie con el transformador.

El modo de operación ZVS, o *Soft-Switching* se alcanza cuando la energía acumulada en la inductancia de fuga del transformador es mayor a la capacidad de la salida de los MOSFET, para que cuando el transistor de una de las “patas” del convertidor se apague, se descargue la salida.

Para extender este modo de trabajo en cargas más ligeras (Entre el 30-50% de la potencia nominal), se añade una bobina  $L_r$  formando un tanque de resonancia con la capacidad de salida de los MOSFET ( $C_{oss}$ ) y el transformador. Por debajo de dichas cargas, el convertidor sigue trabajando en modo *Hard-Switching*. Al conmutar de una polaridad a otra, se dan cambios bruscos en la corriente del primario causando muchos armónicos e incrementando las pérdidas en los transistores.

Este tanque se encarga de que la corriente en el transformador no cambie bruscamente, si no que acomode una pendiente en los estados en los que los transistores están apagados.

Operación y formas de onda

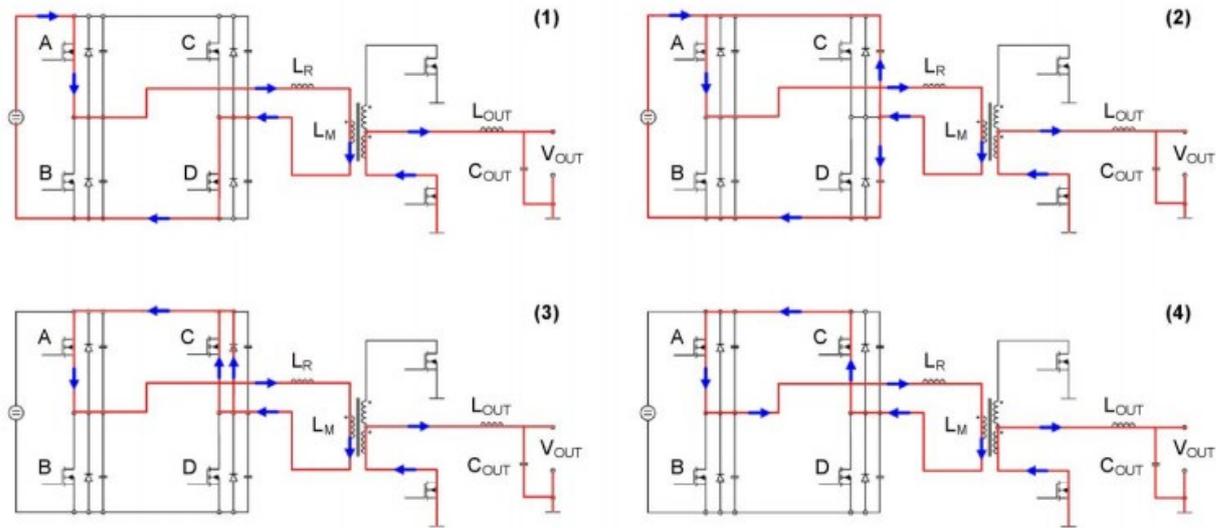


Ilustración 22 - Fase 1 Operación convertidor Full Bridge. Fuente: ZVS Phase Shift Full Bridge CFD2 Optimized Design. Infineon

- Etapa 1. Transmisión de energía: MOSFETs A y D, se encienden y la corriente circula con polaridad positiva por el transformador, que va ascendiendo de acuerdo a su inductancia.
- Etapa 2. Zero Voltage Crossing del MOSFET C: se usa la energía acumulada en la inductancia de resonancia para descargar la capacidad de salida del MOSFET C y cargar la del MOSFET D.
- Etapa 3. Después de que se descargue la capacidad del MOSFET C, la corriente circula por el diodo interno en su cuerpo.
- Etapa 4. Fase de "freewheeling": El MOSFET C se enciende y la corriente circula a través de su canal N y no del diodo.

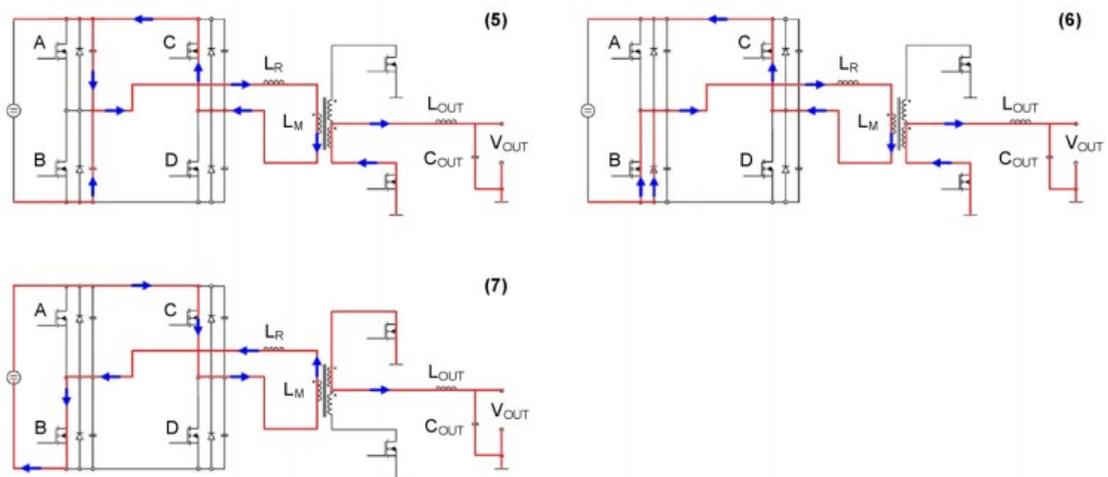
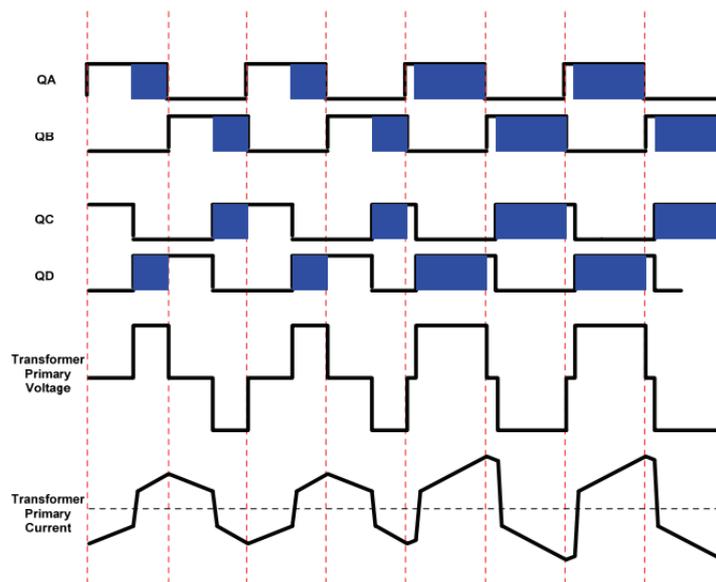


Ilustración 23 - Fase 2 Operación convertidor Full Bridge. Fuente: ZVS Phase Shift Full Bridge CFD2 Optimized Design. Infineon

- Etapa 5. Para comenzar de nuevo la etapa 1, se enciende el MOSFET B. Esta fase se obtiene de la misma manera que la etapa 2, apagando el MOSFET A. La capacidad del MOSFET A se carga al descargar la capacidad del MOSFET B.
- Etapa 6. Se produce la circulación de la corriente del MOSFET B a través del diodo interno.
- Etapa 7. El MOSFET B está encendido, la corriente cambia su dirección y el ciclo empieza de nuevo.

En la siguiente imagen se aprecia cómo cambian la tensión y la corriente en el primario del transformador según se “desplazan” las señales de la segunda “pata” del convertidor. A este modo de control se le llama *Phase Shift*:



*Ilustración 24 - Señales de control convertidor Full Bridge. Fuente: Phase-Shifted Full Bridge DC/DC Power Converter Design Guide. Texas Instruments.*

### 2.4.5. IGBT vs MOSFET

En las fuentes de alimentación de alta potencia, suele aparecer como alternativa, basar el diseño en transistores MOSFET o IGBT. Ambos comparten la misma estrategia de control por tensión en sus puertas, pero su funcionamiento es muy diferente.

Los IGBT's representan una tecnología que se basa en un sustrato de tipo N en la puerta o base, que maneja un transistor de tipo PNP. Esto hace que presentaran inconvenientes, como que fueran propensos a cerrarse inesperadamente, o que resultaran difíciles de colocar en paralelo con otros, así como un coeficiente de temperatura negativo. Aunque estos problemas se han solucionado en la actualidad.

Una de las ventajas que presentan es poder manejar grandes tensiones (>1000V) y grandes corrientes con facilidad, a costa de no poder conmutar a frecuencias elevadas, aumentando el coste de los componentes magnéticos.

Los MOSFET, por otro lado, son una tecnología madura que no presenta grandes problemas en la conmutación y que permite alcanzar frecuencias mucho mayores, pero que no puede resistir grandes tensiones ni corrientes sin tener un coste más elevado.

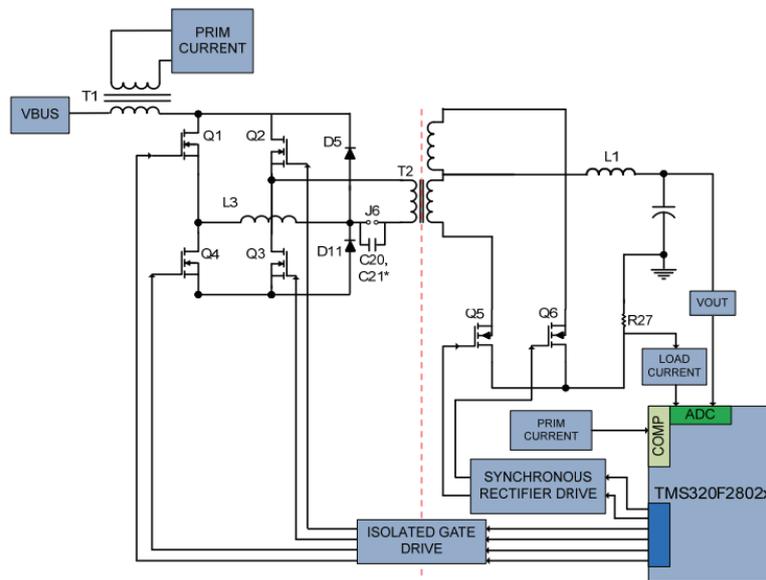
Es por ello, que en diseños de 5KW o más, es muy común encontrar IGBT's, y MOSFET en diseños de menor potencia. En sistemas en los que la potencia no es muy grande y la eficiencia y coste son primordiales, se instalan MOSFET, que permiten reducir el tamaño de los elementos más costosos, como el transformador.

En este trabajo se han utilizado transistores MOSFET ya que se persigue una frecuencia de conmutación de 100KHz para reducir lo máximo posible el tamaño y coste del diseño. Tampoco se requieren grandes corrientes y la tensión a la entrada es lo bastante bajo como para no disparar la  $R_{ds(ON)}$  con MOSFET de alta tensión; lo que afectaría a la eficiencia.

### 2.4.6. Rectificación Activa

En la explicación anterior, se realiza el rectificado de la corriente en el secundario a través de diodos convencionales. A partir de los años 90, la demanda de eficiencia de las fuentes de alimentación hizo que se popularizara un tipo de rectificado basado en transistores MOSFET, llamado rectificación activa o rectificación síncrona, dado que se activan las puertas de los transistores sincronizadas con ciertos eventos.

Es un método que, aunque mucho más complejo, tiene ventajas que lo justifican. En un puente completo, la integración de este sistema se haría como se ve en la imagen:



*Ilustración 25 - Estructura de control Full Bridge y rectificación Síncrona. Fuente: Phase-Shifted Full Bridge DC/DC Power Converter Design Guide. Texas Instruments.*

### 2.4.7. Diodos vs Rectificador Síncrono

Una vez explicado en que consiste la rectificación activa, nos podemos preguntar qué ventajas tiene frente a un rectificado con diodos convencionales, ya que, para empezar, partimos de que tenemos que dedicar recursos a su control.

En diseños en los que se requiera una gran cantidad de corriente, nos vamos a encontrar con unas pérdidas muy altas en forma de calor al trabajar con diodos convencionales. Por ejemplo, suponemos una corriente de 20A en el secundario de nuestro convertidor y un diodo MUR3060PT con una tensión de conducción de 1,2V:

$$\text{Pérdidas en conducción} = 1,2V \times 20A = 24 W$$

Esto nos obliga a tener una solución de refrigeración de mayor calidad y coste, además de que afecta enormemente a la eficiencia global de nuestro sistema.

Con un rectificador síncrono basado en el MOSFET IRFP90N20D, que tiene una Rds (ON) de 0,023 ohmios, podríamos hacer el mismo cálculo:

$$\text{Pérdidas en conducción} = 0,023\Omega \times 20^2 = 9,2 W$$

Como se puede comprobar, las pérdidas en forma de calor se dividen a la mitad, mejorando la eficiencia global del sistema y la temperatura ambiente, haciendo que todo el sistema funcione mejor. Además, tiene un coste sustancialmente menor, ya que reduce el tamaño de los disipadores, uno de los elementos más costosos en estos sistemas.

A continuación, se muestra una tabla comparativa de las pérdidas entre diodos convencionales y dos MOSFET de baja Rds (ON). Se puede apreciar, como hasta los 75A se mantiene la ventaja de usar una rectificación síncrona.

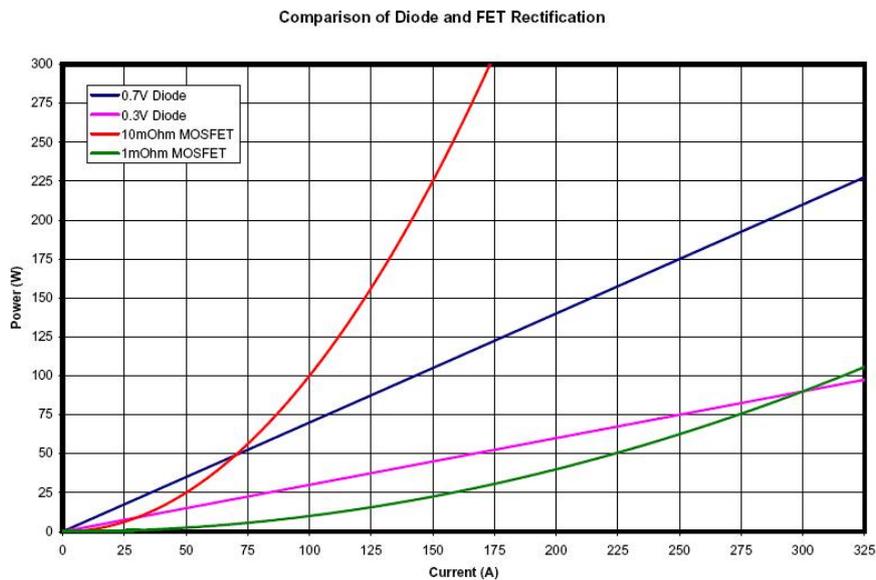
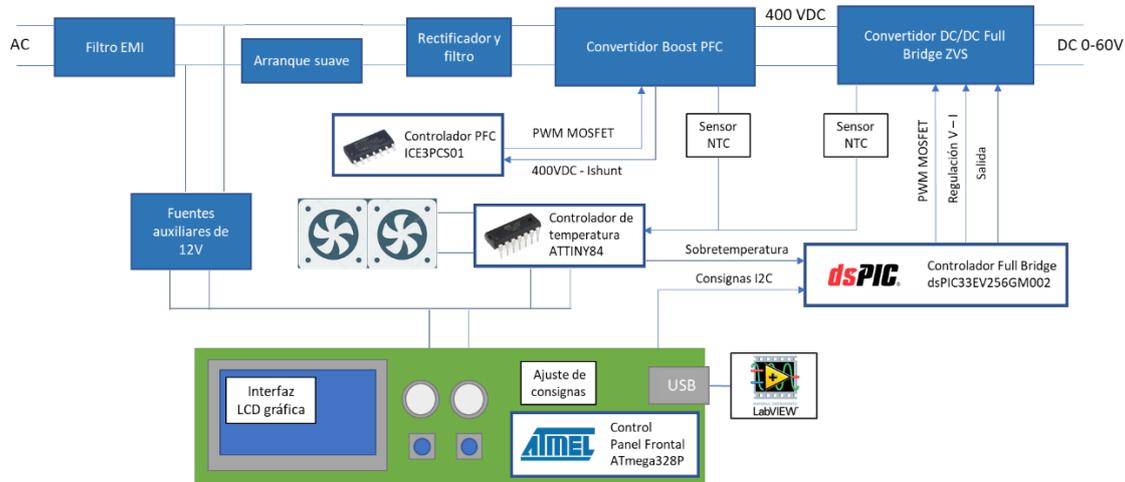


Ilustración 26 - Diodos VS MOSFET en la rectificación del secundario. Fuente: Wikipedia

### 3. Requisitos de diseño



*Ilustración 27 - Esquema de operación de la fuente de alimentación desarrollada en este trabajo*

En este proyecto, se buscará obtener unas prestaciones propias de una fuente comercial de laboratorio, entre las que destacan:

Tensión	Ajustable	0-60V Resolución: 0.02V
Corriente	Ajustable	0-16A Resolución: 0.1A
Potencia	Ajustable	960W
Modos de operación	CV/CC	

Se desarrollará un sistema de corrección del factor de potencia basado en un convertidor elevador de tensión hasta 400V que establezca un factor de potencia unidad en todo el rango de la fuente.

La regulación en corriente continua de la fuente de alimentación se realizará con un convertidor aislado de puente completo ZVS con rectificación síncrona a la salida. El objetivo es alcanzar una eficiencia de un 90% aproximadamente.

El control del convertidor se llevará a cabo con un lazo de control programado en un microcontrolador dsPIC33EV256GM002, que deberá variar el desfase de las señales de control de los transistores en base a una consigna dada.

Las consignas, serán programadas por el usuario mediante una botonera, que será leída por un microcontrolador Atmel Atmega328p. Éste enviará la consigna seleccionada al microcontrolador del convertidor mediante el protocolo de comunicaciones I2C.

Dicho procesador Atmel, será capaz de recibir los resultados de las lecturas del controlador, y deberá mostrarlos en pantalla. Mediante un puerto USB, y un conversor CH340G de UART a USB, enviará dichas lecturas a un ordenador mediante un programa de PC basado en LabView.

En el programa, el usuario será capaz de establecer las consignas y mandárselas a la fuente de alimentación, así como activar una representación gráfica de los datos

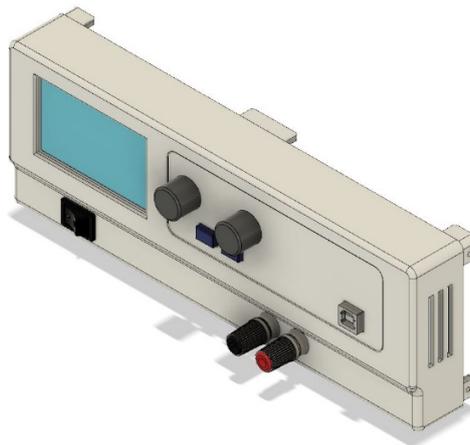
recogidos en la salida. Si se desea, se podrán guardar dichos datos en un archivo de texto.

El usuario también será capaz de programar la carga de una batería, estableciendo la tensión de carga y la tecnología de la batería. Una vez seleccionado, la carga se realizará de manera automática.

## 4. Descripción de soluciones e implementaciones

Se detallan a continuación y consideraciones de diseño de las partes necesarias para el desarrollo del proyecto. En cuanto a los cálculos detallados de las mismas, pueden consultarse en el anexo correspondiente.

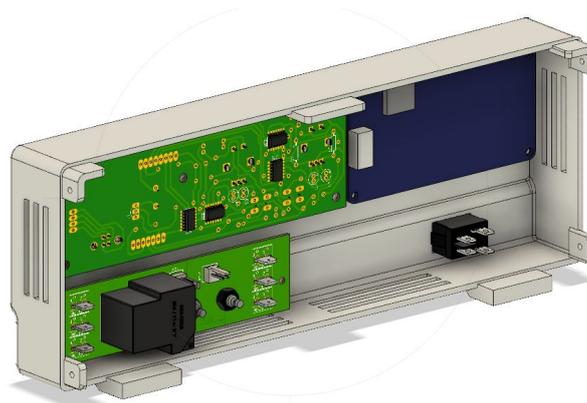
### 4.1. Panel frontal y botones de control



*Ilustración 28 - Modelo 3D Panel frontal. Vista exterior*

Para el ajuste de las consignas y el control de la salida, se ha creado un panel para la parte frontal de la fuente, que sirve de soporte para el interruptor de encendido, los terminales de la fuente, una botonera, un puerto USB, así como una pequeña pantalla.

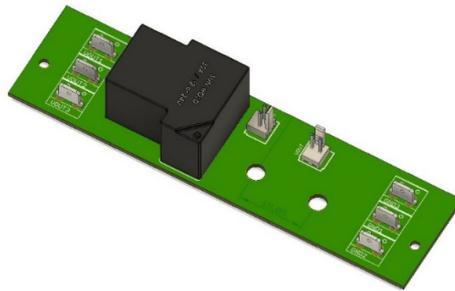
Se han realizado unas hendiduras en los laterales y en la parte inferior para garantizar el flujo de aire a los ventiladores, que aspiraran hacia el interior de la fuente.



*Ilustración 29 - Modelo 3D Panel frontal. Vista interior*

En la parte interior del panel, se han diseñado agujeros para el montaje de las placas de los dispositivos anteriores. El panel debe integrar la etapa de salida de la fuente, la botonera de control, la pantalla y el interruptor de encendido y apagado.

#### 4.1.1. Etapa de salida



*Ilustración 30 - Modelo 3D Etapa de salida*

En esta placa se incluye un relé para que el controlador de la fuente pueda cortar la salida en caso de avería o sobretensión. También permite simplemente, tener la salida apagada mientras ajustamos las consignas en pantalla.

Incluye las conexiones para los terminales de la fuente, así como las clavijas de activación del relé y la lectura de tensión en la salida. Hacer la lectura en este punto permite compensar la caída de tensión en el cableado interno.

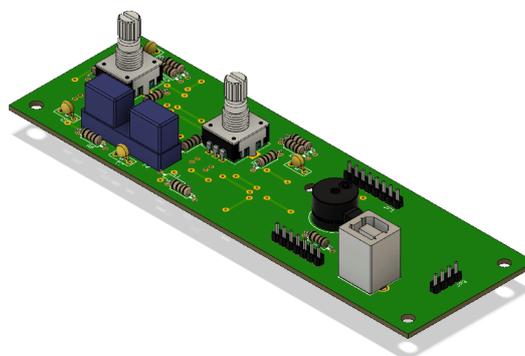
#### 4.1.2. Etapa de salida – Conectores 4 mm



*Ilustración 31 - Terminal de 4 mm "banana" de 30A. Fuente: Aliexpress*

Para realizar la conexión a la fuente de alimentación del dispositivo a probar, se han usado unos conectores de 4 mm, que soportan una corriente de 30A. Se montan sobre la etapa de salida con dos tuercas y se apoyan sobre el panel frontal para mayor rigidez.

#### 4.1.3. Botones de control de la fuente y antirrebote hardware

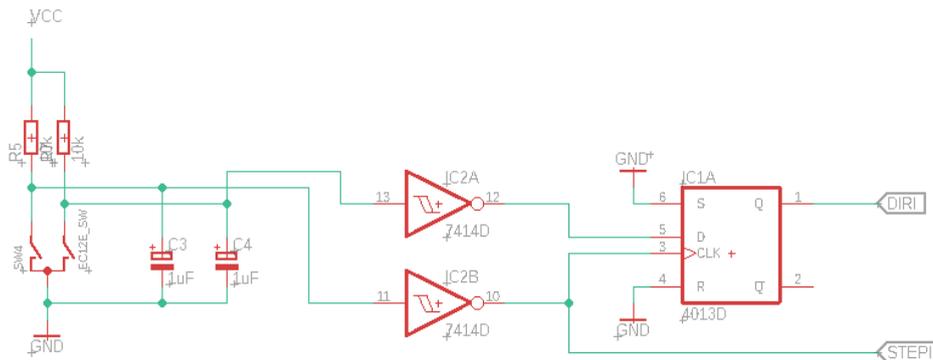


*Ilustración 32 - Modelo 3D Placa de botones y encoders*

La botonera de la fuente está formada por dos encoders rotatorios que permiten ajustar las consignas de tensión y corriente. Dispone de dos botones, un botón de función llamado BFUNC que permite ajustar el incremento en el que cambiamos las consignas en unidades, décimas y centésimas. El botón restante, llamado BOUT activa o desactiva la salida cuando es pulsado.

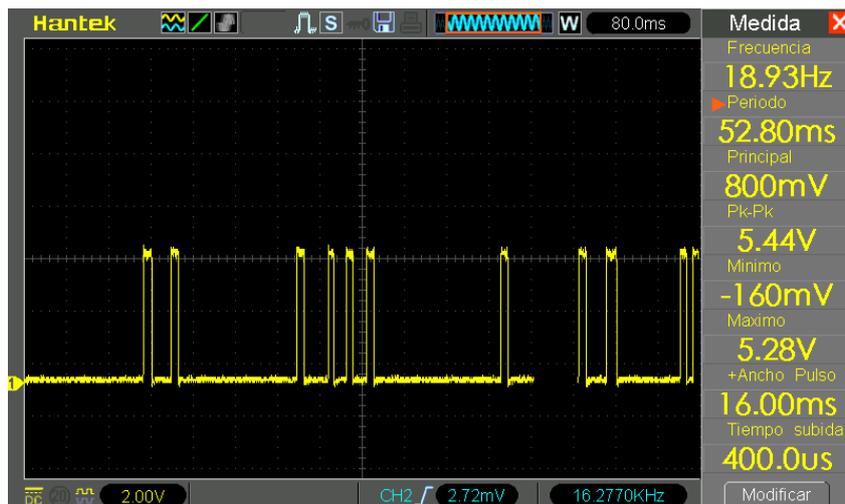
Se integran en esta placa un pequeño altavoz, un puerto USB, y pines de conexión que sirven para insertar la placa de gestión de la interfaz.

Para leer los botones y encoders con mayor solvencia, se ha diseñado un antirrebote hardware. Tiene la ventaja de que ya no nos tenemos que preocupar de integrarlo en el software y es más eficaz. Está formado por un inversor Smith 74HC14 que se conecta a un contador CD4013D.



*Ilustración 33 - Esquema eléctrico antirrebote de encoders*

El resultado que se obtiene de este circuito, es pasar de unas señales diferenciales del encoder, a una unipolar en forma de pulsos en la salida Q del CD4013D. y otra salida en Q negada que cambia de nivel según el sentido de giro del encoder.



*Ilustración 34 - Cadena de pulsos generada por los emcoders en el contador CD4013*



Ilustración 35 - Nivel de tensión generado por el contador CD4013 al cambiar de sentido de giro.

Para realizar un antirrebote en el botón integrado en el encoder, se ha utilizado un filtro de paso bajo que se conecta a un inversor Smith.

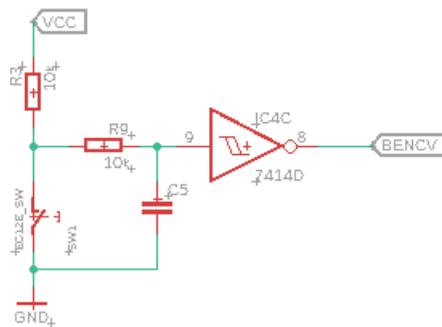


Ilustración 36- Circuito implementado para amortiguar las señales de los botones de los encoders

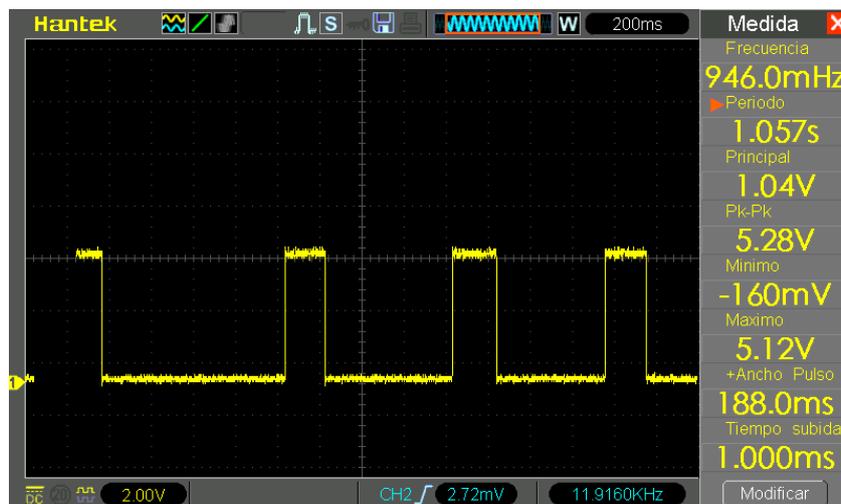
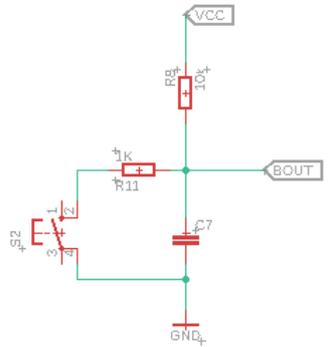
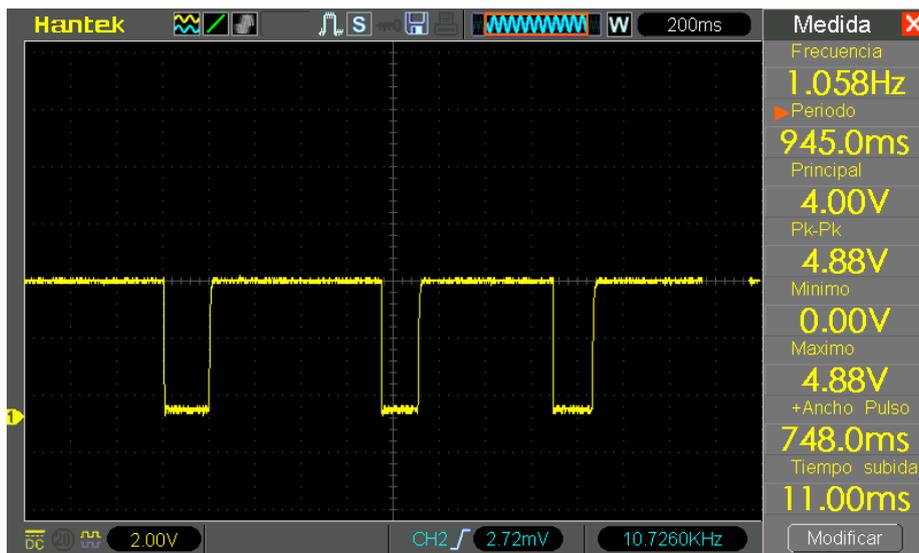


Ilustración 37 - Pulsaciones del botón integrado en el encoder

Debido al poco espacio disponible en la placa, para los botones BFUNC y BOUT, se ha diseñado un antirrebote más modesto, aunque idéntico en funcionalidad, que el de los botones del encoder. Se compone únicamente del filtro de paso bajo y no se ha llevado al inversor, sino que se conecta directamente al microcontrolador.

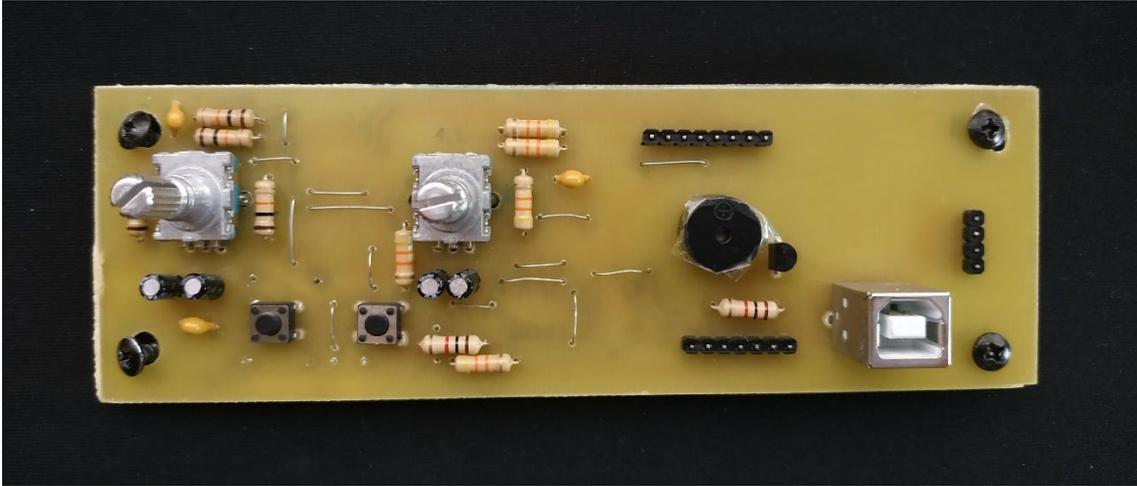


*Ilustración 38 - Implementación antirrebote botones BFUNC y BOUT*

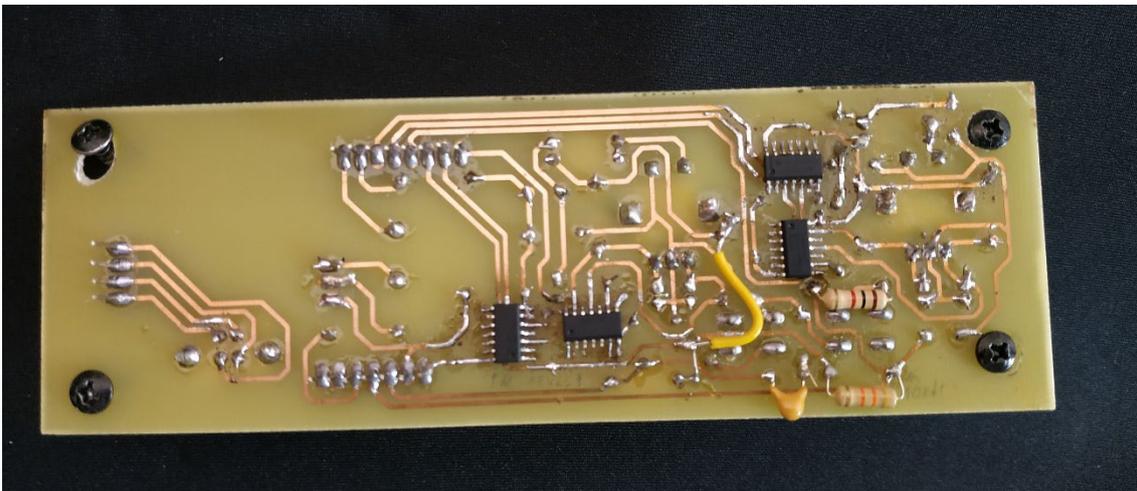


*Ilustración 39 - Pulsaciones generadas por los botones BFUNC y BOUT, nótese que generan un nivel de tensión bajo*

#### 4.1.4. Circuito Impreso



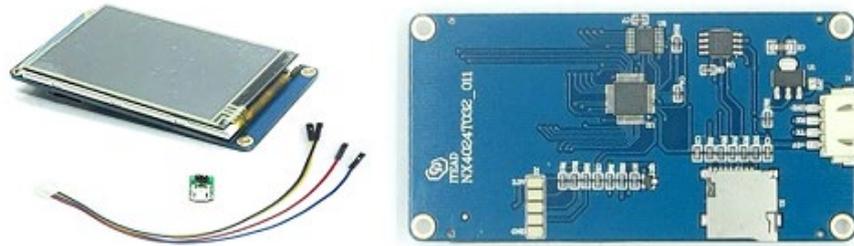
*Ilustración 40 - PCB completa de los botones y encoders. Vista superior*



*Ilustración 41 - PCB completa de botones y encoders. Vista inferior*

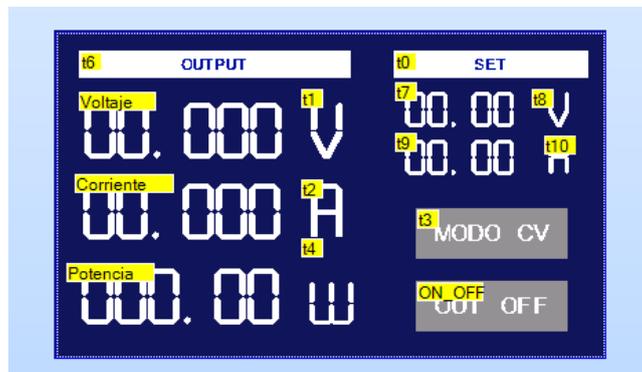
#### 4.1.5. Pantalla Nextion HMI

Para la visualización de consignas y lecturas de tensión, corriente y potencia, además del estado de la fuente, se ha elegido una pantalla Nextion NX4024T032. La gama de pantallas Nextion, se basan en STM32 y mediante comunicación serie UART. Actúan como una interfaz entre la pantalla y nuestro sistema, que se comunican por comandos.



*Ilustración 42 - Pantalla Nextion 3.2". Vista superior e inferior*

Este método, nos permite descargar al procesador de nuestro sistema de las tareas de pintado de la pantalla, especialmente al tratarse de pantallas gráficas. Otra ventaja, es poder diseñar la interfaz que queramos desde un editor bajo PC, llamado Nextion Editor. Con esta herramienta podemos diseñar, escribir código y simularlo antes de cargarlo en la pantalla.



*Ilustración 43 - Interfaz diseñada en Nextion Editor*

Estas pantallas están equipadas con una ranura microSD para su programación, aunque también admiten ser programadas desde un adaptador USB a RS232.

#### 4.1.6. Interruptor AC

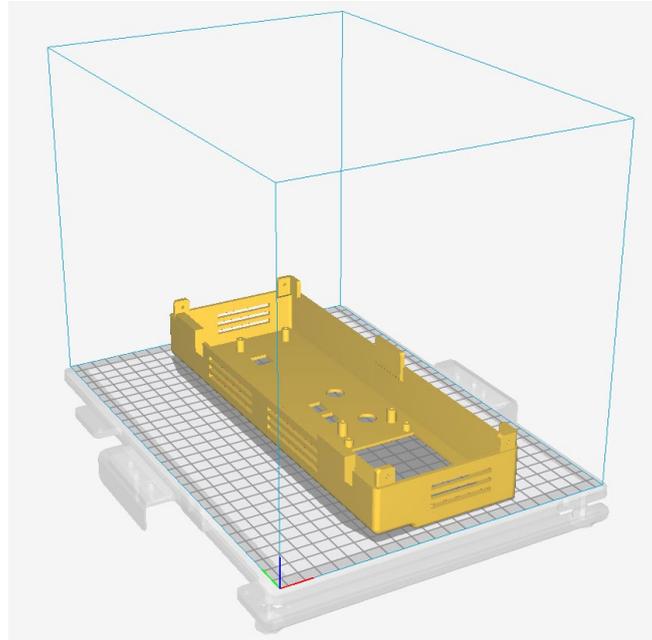


*Ilustración 44- Interruptor AC 10A 21x15 mm. Fuente: Aliexpress*

Se ha utilizado un interruptor AC de 10A de 21 x15 mm para controlar la conexión a la red y poder apagar y encender la fuente por completo.

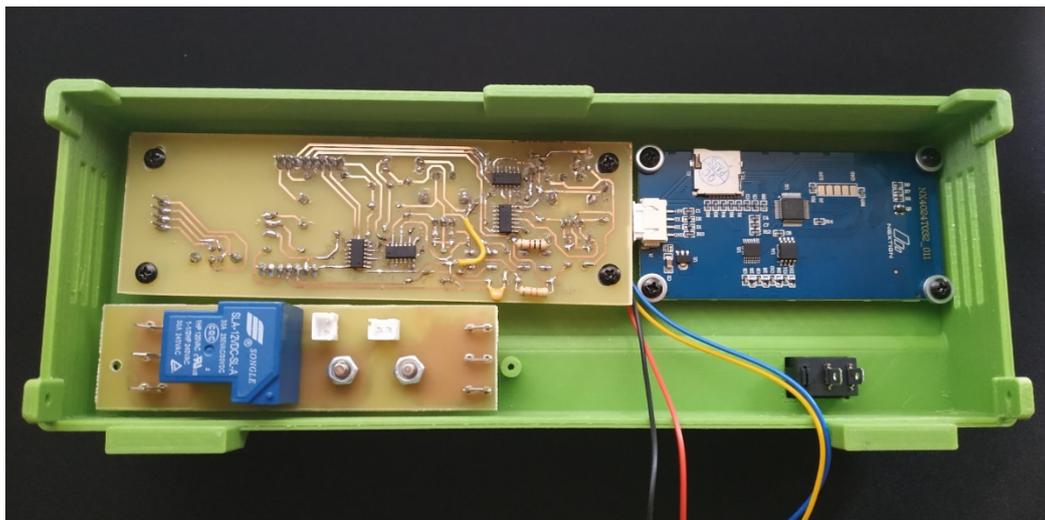
#### 4.1.7. Construcción

Para la construcción del panel frontal, se ha utilizado una impresora 3D y material de impresión PLA. Un relleno a 20% y 1,2mm de pared exterior. El resultado es una pieza prácticamente maciza con buena rigidez y resistencia mecánica.

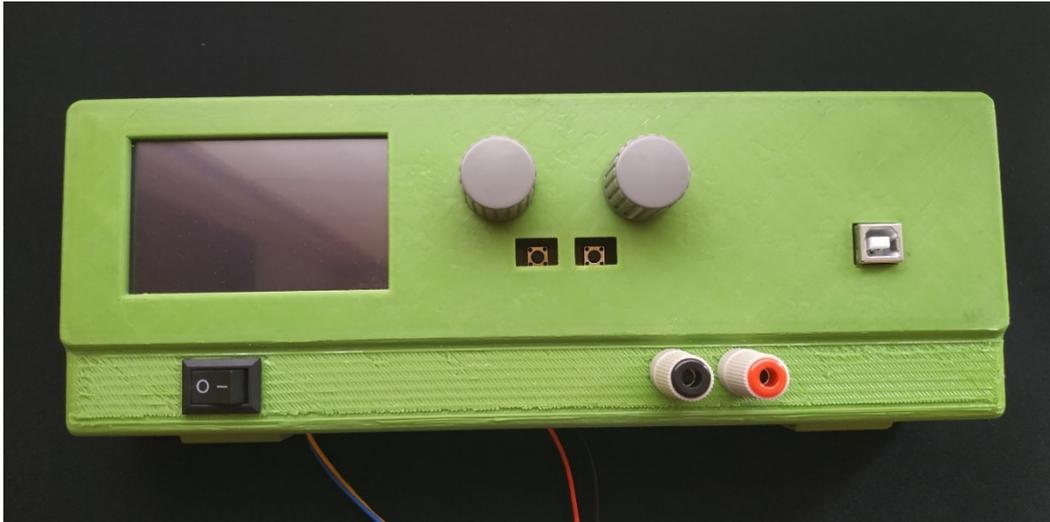


*Ilustración 45 - Vista en Cura 3D del panel frontal en la cama de BQ Hephestos 2*

El tiempo de impresión de la pieza es de, aproximadamente, doce horas y media, utilizando un total de 116 gramos de plástico.

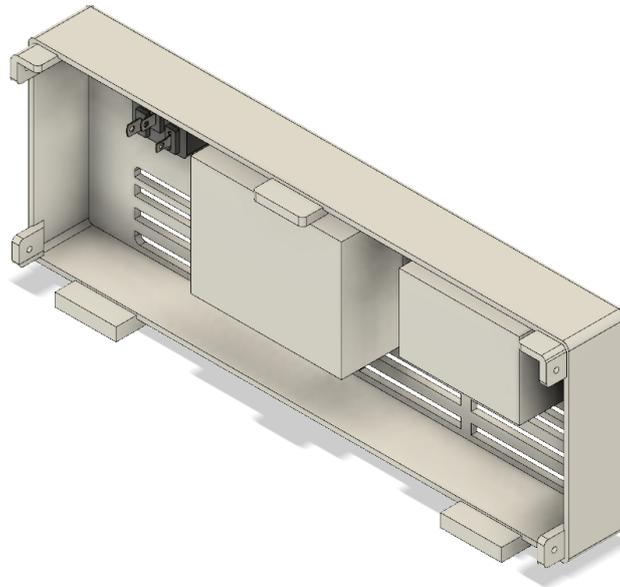


*Ilustración 46 - Montaje final del panel frontal con la placa de botones y encoders, la etapa de salida y la pantalla. Vista interior*



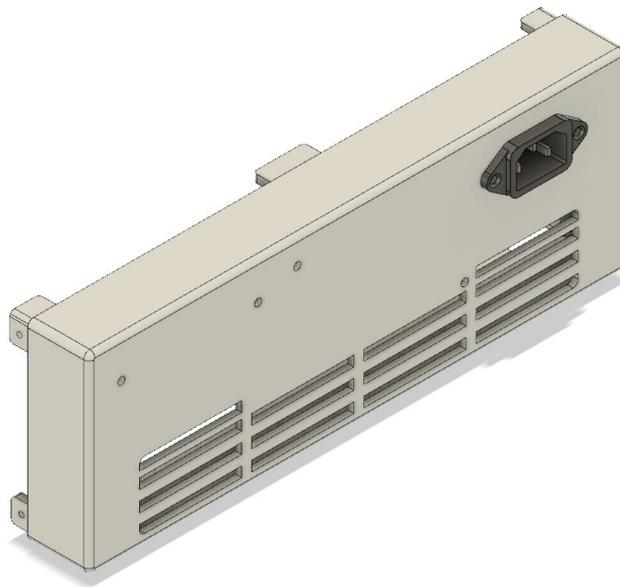
*Ilustración 47. Montaje final panel frontal con cubrebotones, interruptor AC, terminales 4 mm y puerto USB. Vista exterior*

## 4.2. Panel trasero



*Ilustración 48 - Modelo 3D Panel trasero con fuentes de alimentación. Vista interior*

Para colocar las fuentes de alimentación auxiliares del sistema y la toma de corriente, se ha diseñado un panel que irá colocado en la parte posterior de la fuente. También se han colocado unas hendiduras para que el aire que aspiran los ventiladores de la parte delantera pueda salir por detrás.



*Ilustración 49 - Panel trasero. Vista exterior*

### 4.2.1. Conector AC



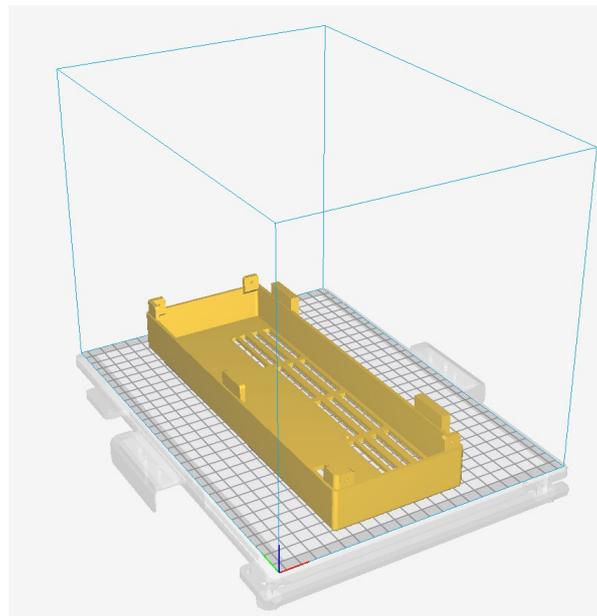
*Ilustración 50 - Conector AC 10ª. Fuente: Aliexpress*

Para la entrada de corriente alterna a la fuente de alimentación, se ha utilizado un conector de tres terminales, para poder referenciar el dispositivo a masa, con capacidad para 10A.

#### 4.2.2. Fabricación

Para su construcción se ha utilizado una impresora 3D, con material de impresión PLA, una pared exterior de 1,2mm y un relleno al 20%. El resultado es una pieza prácticamente maciza con buena rigidez y resistencia mecánica.

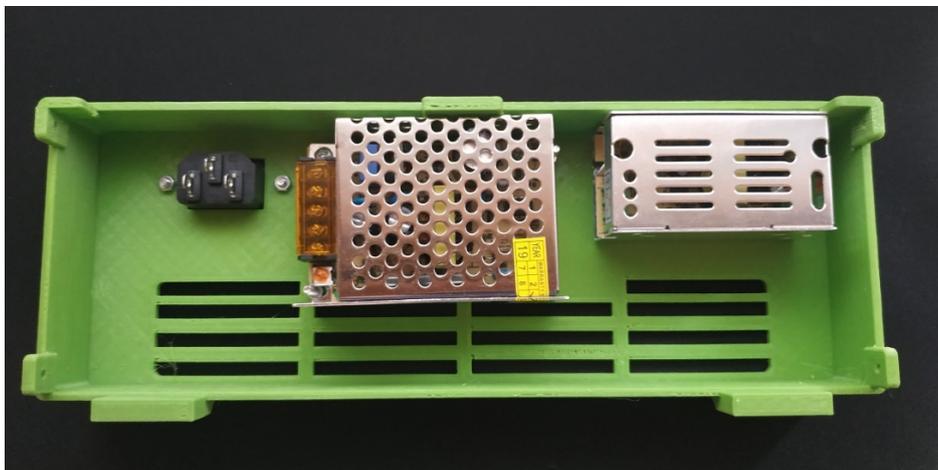
El tiempo de impresión de la pieza es de, aproximadamente, once horas y media, utilizando un total de 109 gramos de plástico.



*Ilustración 51 - Vista en Cura 3D del panel trasero en la cama de BQ Hephastos 2*



*Ilustración 52 - Montaje panel trasero con conector AC. Vista exterior*

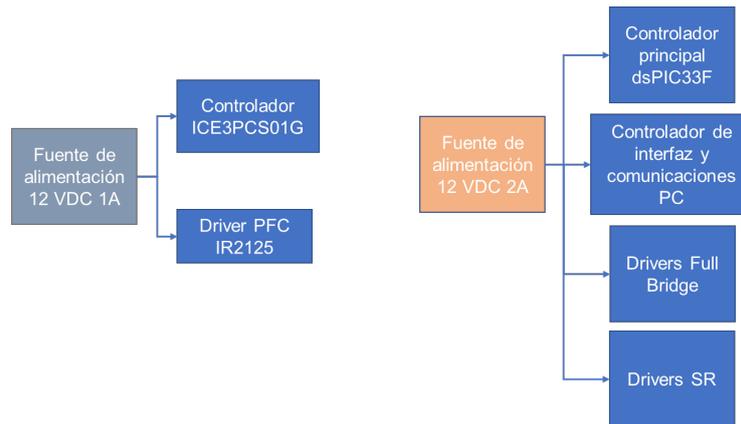


*Ilustración 53 - Montaje panel trasero con fuentes de alimentación. Vista interior*

#### 4.2.3. Fuentes auxiliares

El uso de fuentes de alimentación auxiliares es necesario para que funcionen otros elementos del sistema que se encargan de controlar el proceso de conversión de la energía.

Los sistemas que necesitan dicha energía son: el controlador del convertidor PFC, el controlador del convertidor DC-DC y sus drivers, el sistema de gestión de interfaz, y el sistema de control de temperatura y ventiladores.



*Ilustración 54 - Esquema básico de alimentación de los circuitos auxiliares*

Dado que el controlador PFC se encuentra en el lado de alta tensión, necesita una fuente auxiliar a parte, que lo separe de la interfaz y el controlador principal, que se encuentran en el lado de baja tensión.

Se han utilizado dos fuentes genéricas de 12V, una con capacidad de 1A y 12W para el controlador PFC, y otra con capacidad de 2A y 24W, para el controlador principal, la interfaz y el control de ventiladores.



*Ilustración 55 - Fuentes de alimentación utilizadas. Izquierda: fuente de 12V 1A. Derecha: fuente de 12V 2A. Fuente: Aliexpress*

#### 4.2.4. Posibilidad de mejora

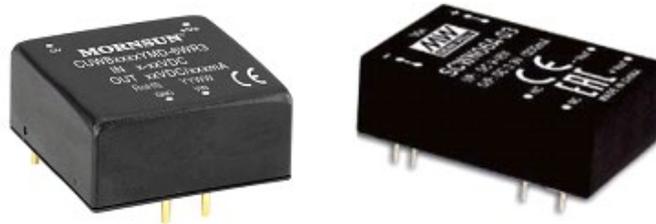
Para este trabajo, por cuestión de costes, se han instalado dos fuentes de alimentación diferentes para los dos controladores. No es la única solución disponible, ya que hay otra que ofrece una serie de ventajas.

El hecho de tener dos dispositivos más emitiendo armónicos aparte del convertidor principal, reduce el factor de potencia. Cuando se alcance cierto nivel de potencia, sus interferencias serán pequeñas en comparación, pero a baja potencia, puede influir notablemente.

Es por ello, que se puede reducir el número de fuentes de alimentación adicionales a una sola, utilizando un convertidor DC-DC aislado para el controlador PFC. Dado el bajo consumo de esta aplicación, el convertidor a utilizar no necesita ser muy voluminoso.

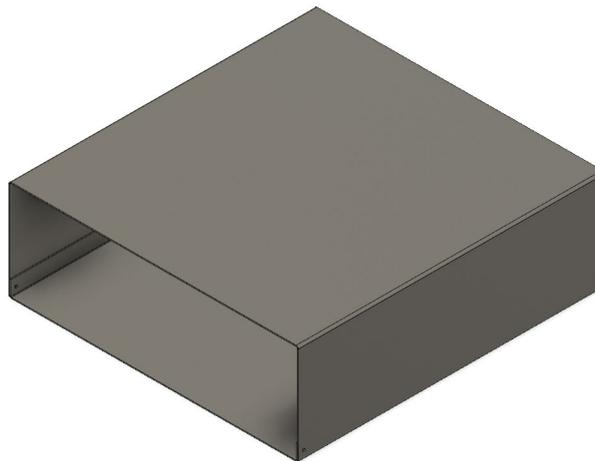
Se proponen dos alternativas de bajo coste, la primera, un convertidor del fabricante MORSUN modelo CUWB1212YMD-6WR3 con una tensión de entrada entre 4.5 y 36V y una salida de 12V. La tensión máxima de aislamiento es de 1.5KV.

Como segunda opción, se propone un convertidor del fabricante MEAN WELL, modelo DCWN06A-12 con una tensión de entrada entre 9 y 12V, y una salida de 12V. La tensión máxima de aislamiento es de 3KV.



*Ilustración 56 - Convertidores aislados como alternativa para el controlador PFC. Izquierda: MORSUN. Derecha: MEAN WELL*

#### 4.2.5. Cuerpo de la fuente de alimentación



*Ilustración 57 - Modelo 3D caja exterior de acero de la fuente de alimentación*

A modo de plataforma para colocar las placas de circuitos de la fuente de alimentación, así como para unir el panel frontal y el trasero, se han diseñado dos láminas de acero de 1 mm de espesor en forma de U.

Se ha elegido acero como material ya que aporta gran rigidez a la estructura. Como contrapartida es más difícil de trabajar que el aluminio y tiene un peso mayor.

#### 4.2.6. Lámina inferior

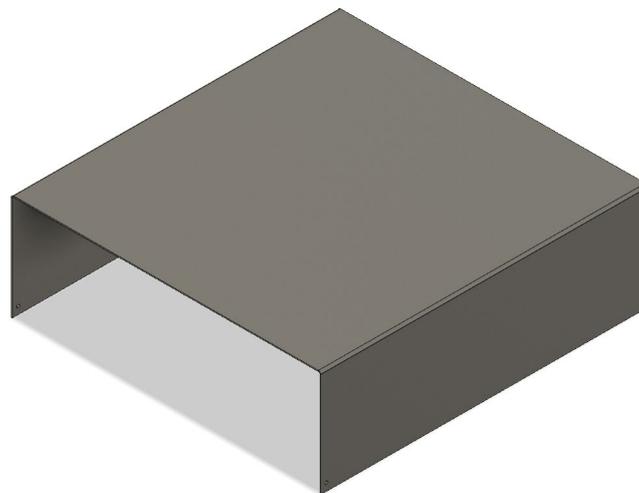


*Ilustración 58 – Modelo 3D Lámina de acero inferior*

La base, consiste en una lámina de dimensiones 280x290 mm que se dobla con unos pliegues a los laterales de 14 mm, para alcanzar unas dimensiones finales de 280x265 mm.

Esta primera lámina sirve de apoyo para las placas de circuito impreso del convertidor PFC y del convertidor DC-DC de puente completo. También se conecta a esta lámina la conexión a tierra.

#### 4.2.7. Cubierta superior



*Ilustración 59 - Modelo 3D Lámina de acero superior*

La misión de esta segunda lámina es cubrir la fuente y portar rigidez a la estructura. Conecta el panel frontal y el trasero mediante dos tornillos y evita que se doblen hacia atrás al manipular los botones o cables.

También evita que las personas puedan acceder al interior de la fuente, que dadas las tensiones que se manejan en el interior, puede resultar en heridas graves.

### 4.3. Sistema de filtrado EMI y convertidor PFC

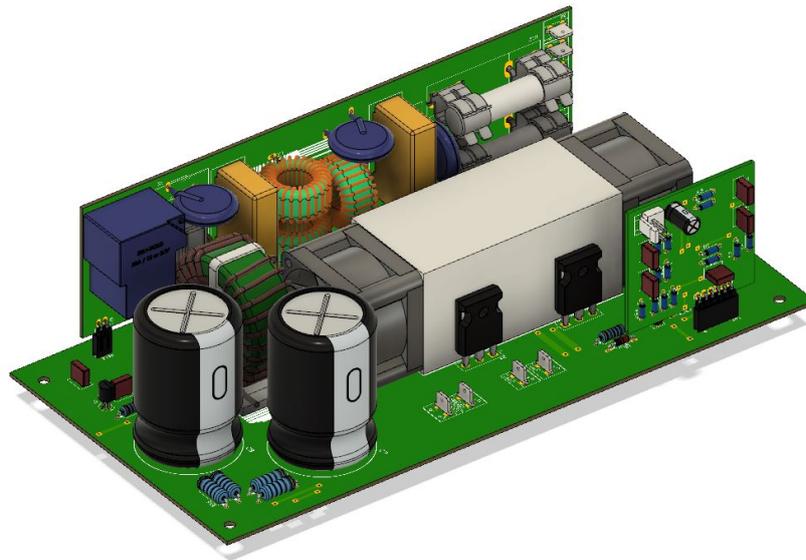


Ilustración 60 - Modelo 3D montaje Filtro EMI y convertidor Boost PFC

Con las consideraciones anteriormente citadas sobre el vertido de armónicos y su influencia en factor de potencia, así como las técnicas empleadas para su corrección, se ha presentado una solución para potencias de hasta 1KW.

Se trata de un conjunto que incluye la protección mediante fusibles a la entrada, un sistema de arranque suave, un filtro EMI, un convertidor elevador de tensión como corrector del factor de potencia y su controlador.

#### 4.3.1. Etapa de filtro EMI y arranque suave

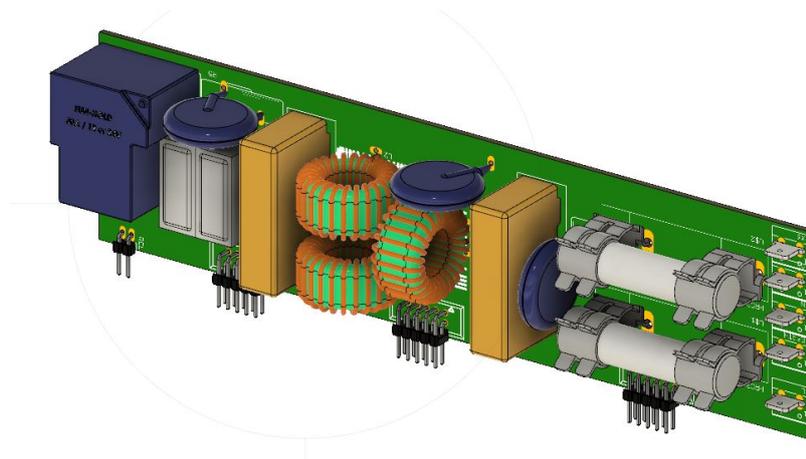


Ilustración 61 - Modelo 3D etapa de entrada y filtro EMI

Se trata de una tarjeta de montaje vertical para ahorrar espacio dentro de la caja. En esta placa se encuentran las protecciones a la entrada, tanto de tensión como de corriente, así como el filtro EMI y el sistema de arranque suave de la fuente.

#### 4.3.2. Elemento de protección contra sobrecorriente: Fusibles

Cuando se diseña un sistema eléctrico que va a demandar potencia de la red, es obligatorio equiparlo con fusibles de protección en caso de avería para evitar daños en los componentes internos del equipo, o incendios en el caso de que algunos componentes alcancen grandes temperaturas.

Dado que el sistema se ha diseñado para 1KW y la tensión mínima que puede llegar en la red eléctrica española es de 207VAC en la red, se han dimensionado los fusibles para corrientes de 10A. A priori, puede parecer que se ha elegido un fusible sobredimensionado, pero veremos que existen corrientes de pico que debe resistir. (Véase en el Anexo – Cálculos *Dimensionamiento de la bobina*)



*Ilustración 62 - Fusible 10A 38 mm HRC. Fuente: Aliexpress*

Los fusibles elegidos son del fabricante CHNT modelo RT14 de dimensiones 10x38 mm y alta capacidad de ruptura. Esto quiere decir, rellenos de arena, para evitar que estallen y se esparzan fragmentos de vidrio dentro del sistema. Se han colocado tanto en la fase como en el neutro.

#### 4.3.3. Elemento de protección contra sobretensiones: Varistor



*Ilustración 63 - Varistor ZOV 20D431k 275VAC. Fuente: Aliexpress*

El siguiente elemento de protección recomendable es el varistor. Un varistor, que se conecta en paralelo a la alimentación, se encarga de entrar en conducción cuando cae entre sus terminales una tensión mayor que la determinada por el fabricante, llamada tensión de disparo. Es un dispositivo con un funcionamiento que muestra ciertas similitudes a la curva de conducción de un diodo; pero con una tensión de codo muy superior, y una característica de polarización simétrica.

El hecho de incorporar un varistor, nos permite evitar que la corriente circule por ciertos elementos sensibles que podrían dañarse por un pico de tensión. Al entrar en conducción el varistor puede autodestruirse, pero es preferible a tener daños mayores.

Se recomienda su colocación inmediatamente después de un fusible. En este caso se ha colocado uno justo después de los fusibles, y dos conectados en fase y neutro, con la tierra de la red.

Se han seleccionado varistores con unas características de disparo de 275 VAC RMS y 350 VDC; modelo 20D431K del fabricante ZOV.

#### 4.3.4. Filtro EMI

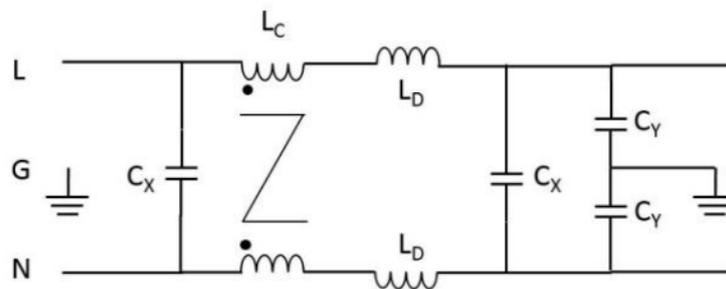


Ilustración 64 - Esquema típico filtro EMI

Para el diseño de nuestro filtro se ha usado el esquema de un filtro en modo común de segundo orden, junto con un filtro en modo diferencial, para intentar atenuar el máximo ruido posible reduciendo los costes.

Se han usado como componentes en modo común, una bobina  $L_C$  de 5 mH y que soporta 10A. Como condensadores  $C_Y$  se ha elegido una capacidad de 4700 pF y 1 KV de tensión nominal.

En una primera aproximación, se estimó una frecuencia de corte de 35 KHz, que resultó en una inductancia necesaria de 2 mH. Al encender el convertidor, inmediatamente se cortaba el suministro eléctrico al saltar el diferencial del cuadro. Se estimó entonces, que la frecuencia de corte era demasiado alta y se vertían a tierra demasiados armónicos que activaban las protecciones, por lo que se rehicieron los cálculos con una frecuencia de corte de 25 KHz. (Véase en el Anexo – Cálculos *Atenuación del ruido en modo común*)

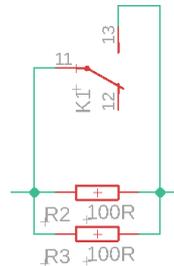
Para el filtrado del ruido en modo diferencial, se ha elegido un condensador  $C_x$  de 1.2 uF y dimensionado para 275 VAC. Para las bobinas de modo diferencial, se han utilizado 100uH de inductancia, con una capacidad de corriente de 10A como la de modo común. (Véase en el Anexo – Cálculos *Atenuación del ruido en modo diferencial*)

#### 4.3.5. Arranque suave

Después del filtro, debemos colocar una etapa de arranque suave. Esto es debido a que, en las fuentes conmutadas, la corriente de *inrush* es muy alta. De esta manera, al encender el sistema, se podría dañar por el pico de corriente o, en el mejor de los casos, hacer saltar el magnetotérmico de la red.

En este diseño, se ha realizado un arranque suave con dos resistencias en paralelo, en serie con la carga, de 100 ohmios y 5W de potencia que evitan que esto ocurra.

Una vez que se ha alcanzado el 95% de la tensión nominal de funcionamiento, la corriente disminuye y se puede operar en modo normal. Es entonces cuando se activa un relé en paralelo con dichas resistencias, para que la corriente deje de circular por ellas y el sistema funcione a plena potencia.



*Ilustración 65 - Esquema del arranque suave implementado*

El relé utilizado es del fabricante SONGLE, modelo SLA-12VDC-SL-A, cuya bobina funciona con 12 VDC y puede resistir hasta 30A.



*Ilustración 66 - Relé SONGLE SLA-12VDC-SL-A. Fuente: Aliexpress*

#### 4.3.6. Terminales de conexión de potencia



*Ilustración 67 - Terminal de crimpar Faston 6.3 mm. Fuente Aliexpress*

Esta tarjeta incluye conexiones con terminales de crimpar Faston de 6.3 mm donde se conectan los cables de entrada de corriente alterna de la red. Se han añadido varios terminales extra en el Neutro para poder conectar las fuentes de alimentación auxiliares.

También se ha dividido la pista de fase hacia un interruptor que se encarga de conmutarlo para poder encender y apagar la fuente. A esta división se le añaden dos terminales más para poder apagar y encender las fuentes de alimentación auxiliares.

Para conectar verticalmente esta placa con el convertidor, se ha usado un conector de tipo Pin Header de 2x5 pines y orientación a 90 grados.

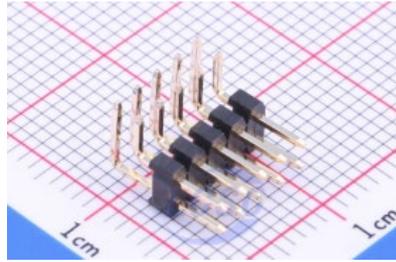


Ilustración 68 - Pin Header 2x5. Fuente: LCSC

Cada uno de los pines, puede resistir una corriente de 3A, al haber diez de ellos, cumple con las demandas de la fuente.

#### 4.3.7. Resultado del circuito impreso

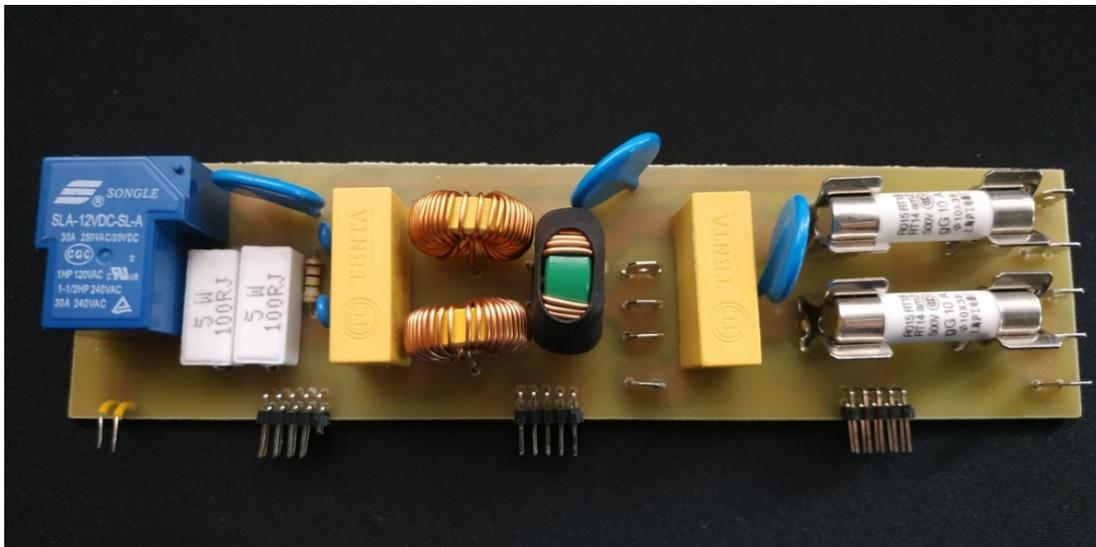


Ilustración 69 - PCB etapa de entrada y filtro EMI. Vista superior

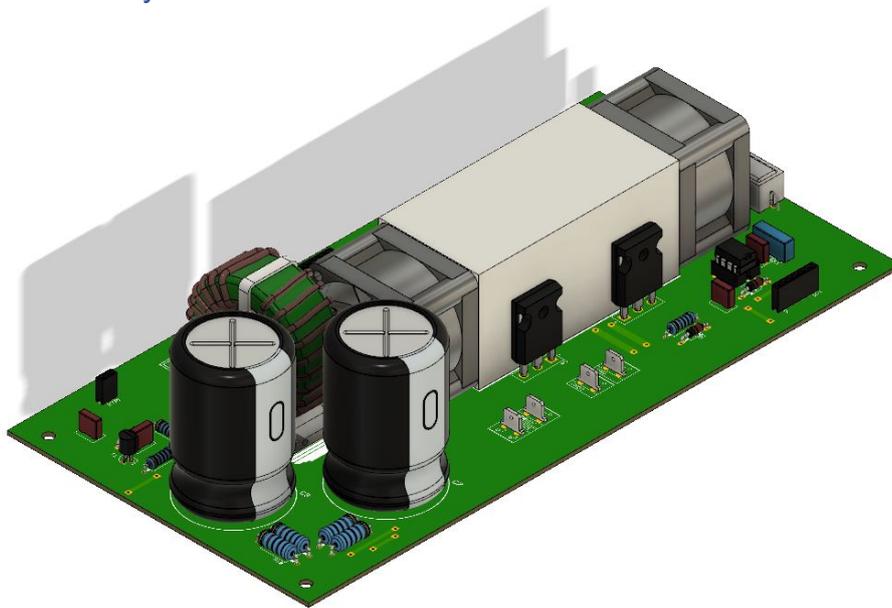


Ilustración 70 - PCB etapa de entrada. vista inferior. Se observan las pistas estañadas

Dado la gran cantidad de corriente que puede circular por este conjunto de placas, se ha recubierto el cobre con estaño. Esto incrementa la capacidad de corriente de las pistas, ya que disminuye su resistencia nominal al colocarse ambos materiales en paralelo.

El hecho de recubrir las pistas con estaño, también elimina la oxidación del cobre al estar en contacto con el exterior. Fenómeno que se ve acelerado por la circulación de corriente, por lo que es recomendable su recubrimiento o pintarlo con un barniz antioxidante.

#### 4.3.8. Rectificado y convertidor PFC



*Ilustración 71 - Modelo 3D convertidor Boost PFC*

Esta placa integra los conectores para montar verticalmente el módulo de arranque suave y filtrado EMI, el rectificador, y el convertidor DC-DC elevador que corrige el factor de potencia, así como los circuitos necesarios para su funcionamiento. También se incluyen los componentes que descargan de manera segura de los condensadores que funcionan a 400V.

Este módulo necesita de refrigeración activa dada la potencia que disipa, como veremos a continuación, por lo que monta el disipador y los ventiladores seleccionados.

### 4.3.9. Rectificador de onda completa

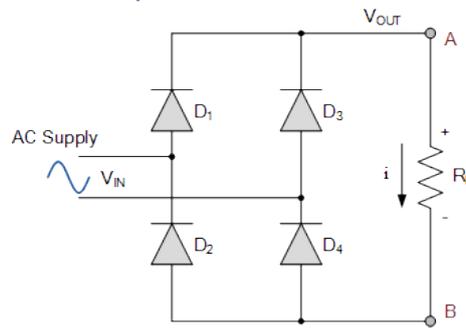


Ilustración 72 - Rectificador de onda completa. Fuente: electronics-tutorials.ws

Para el paso de corriente alterna a corriente continua, se usa un dispositivo llamado rectificador de onda completa. Está formado por 4 diodos conectados en puente, que convierten en tensión continua incluso el medio ciclo negativo de la alterna.

El resultado de esta rectificación es una forma de onda cuasi-continua, es decir, la tensión resultante es positiva, pero la forma de onda es idéntica a la alterna, solo que se invierten los picos negativos. Se puede apreciar mejor en la siguiente imagen:

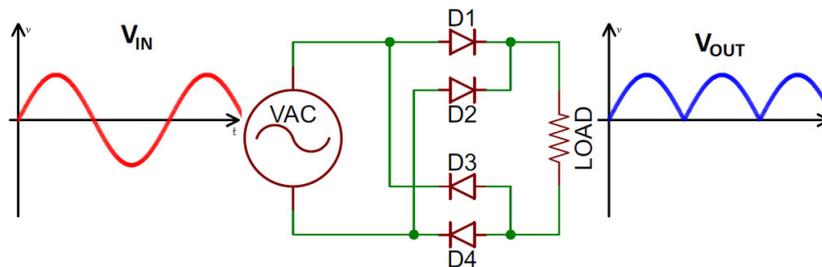


Ilustración 73 - Vista de la onda rectificada sin filtrar. Fuente: elprocus.com

Con este circuito, se obtiene una forma de onda que tiene el doble de frecuencia que la onda original. Es decir, en España, la red proporciona corriente alterna a 50Hz, por lo que se obtiene a la salida una onda que oscila a 100Hz.

Este tipo de señales, están lejos de considerarse corriente continua, por lo que se deben filtrar con un condensador para subir el nivel de tensión llenando los “huecos” entre los valles de la onda.

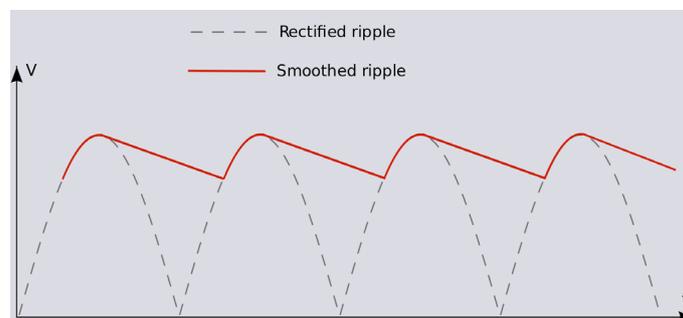


Ilustración 74 - Vista de la onda rectificada con filtro C. Fuente: DiodeLab.com

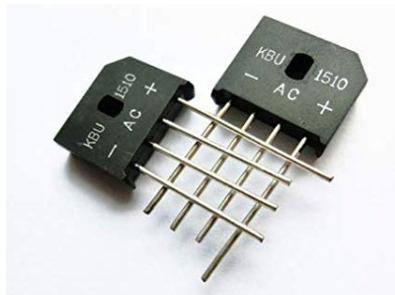
Una vez se ha producido la rectificación completa de la corriente alterna, la tensión final, es la tensión de pico de la tensión alterna. Ya que, pasamos de medir la tensión en

RMS, (propio de la alterna para hacer una equivalencia con valor energético de la continua) a medir tensión continua, donde no necesitamos esta concepción.

Aun así, este tipo de conversión presenta un gran nivel de rizado cuando se demandan corrientes considerables, ya que es un sistema sin regulación. Para mitigar estos efectos, se requieren de grandes bancos de condensadores, que puedan estabilizar la tensión a según qué corrientes.

Se ha elegido un rectificador de onda completa comercial, del fabricante SEP, modelo KBU1510 en base a los requerimientos de corriente y potencia disipada. (Véase en el Anexo – Cálculos *Dimensionamiento del rectificador*)

Es capaz de resistir 15A de manera continua y 300A en picos de 8.3 ms, tiene una tensión de pico repetitiva inversa de 1000V y una caída de tensión en conducción de 1V por diodo.



*Ilustración 75 - Puente de diodos KBU1510. Fuente: Aliexpress*

Dado que nuestro sistema lleva un convertidor DC-DC para corregir el factor de potencia colocado justo después del rectificador, no nos preocupa la tensión de rizado, por lo que no es necesario colocar un gran condensador de filtro en este caso. Solo se ha añadido un condensador de 470nF y 630V para que la tensión sea algo más estable al entrar al convertidor.

4.3.10. Convertidor DC-DC elevador de tensión

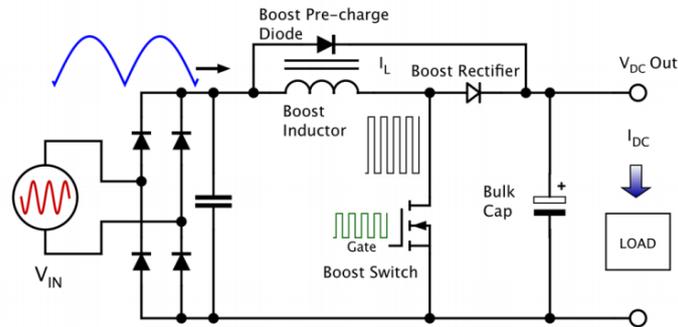


Ilustración 76 - Esquema básico convertidor Boost PFC. Fuente: PFC Boost Converter Design Guide. Infineon

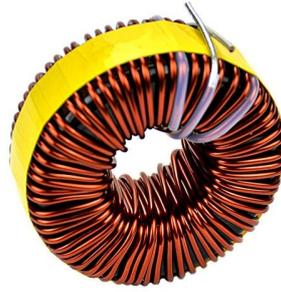
Antes de comenzar el diseño, debemos definir una serie de parámetros que van a determinar las especificaciones sobre las que diseñar nuestro convertidor:

Tensión de entrada mínimo	207 VAC (230 - 10%)
Tensión de entrada máximo	253 VAC (230 + 10%)
Tensión de salida	400 VDC
Potencia	1000 W
Frecuencia de conmutación	100 KHz
Tensión de rizado	10Vpp (400 +-5%)
Eficiencia ( $\eta$ )	0.93
Tiempo Hold-Up	16.6 ms

4.3.11. Bobina PFC

La bobina elegida es proporcionada por el fabricante Aiyima, tiene una inductancia de 2mH y una corriente de saturación de 12A, por lo que cumple con las especificaciones de diseño. Para dimensionar la bobina, se han realizado los cálculos para satisfacer las condiciones de diseño de la red eléctrica española de 230 VAC.

Para cumplir las condiciones de diseño de la bobina de la red eléctrica de 120VAC, se necesitaría una bobina que no se ha podido obtener ya que tenía un precio demasiado alto. Por lo que la fuente de alimentación solo es capaz de entregar la máxima potencia conectada a 230VAC. (Véase en el Anexo – Cálculos *Dimensionamiento de la bobina*)



*Ilustración 77 - Bobina Aiyima 2 mH. Fuente: Aliexpress*

#### 4.3.12. MOSFET Boost PFC

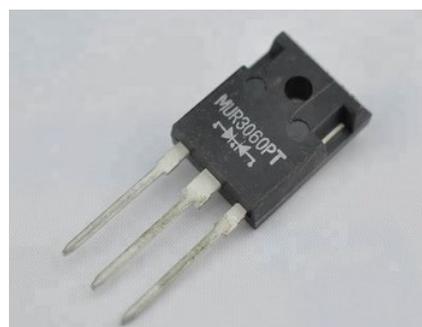
Como transistor MOSFET encargado del funcionamiento del convertidor, se ha elegido una unidad del fabricante VISHAY modelo IRFP460. Como características más importantes destacan su baja  $R_{ds(ON)}$ , de tan solo  $0.27\Omega$  y una tensión de operación máximo de 500V. Es capaz de soportar 13A con una temperatura de unión de  $100^{\circ}\text{C}$ . Como se puede comprobar en su apartado correspondiente en los cálculos, cumple con las condiciones de diseño. (Véase en el Anexo – Cálculos *Dimensionamiento del transistor MOSFET*)



*Ilustración 78 - Transistor MOSFET IRFP460. Fuente: Aliexpress*

#### 4.3.13. Diodo de rectificación Boost

En base a los requerimientos, se ha seleccionado un diodo del fabricante ON SEMICONDUCTOR, modelo MUR3060PT con una caída de tensión en conducción  $V_f$  de 1.2V, capaz de aguantar una corriente directa de 15A y una tensión de pico repetitiva inversa de 600V. (Véase en el Anexo – Cálculos *Dimensionamiento del diodo Boost*)



*Ilustración 79 - Diodo doble MUR3060PT. Fuente: Aliexpress.*

#### 4.3.14. Condensadores de filtro de salida

Para cumplir la especificación del tiempo de mantenimiento de la tensión de 16.66 ms, se eligen dos condensadores electrolíticos EPCOS, modelo B43504A5477M06 de 470uF y una ESR de 0.29Ω a 100Hz. Tienen unas dimensiones de 35x50mm. Al colocarlos en paralelo, se comportan como uno de 940uF. (Véase en el Anexo – Cálculos *Dimensionamiento condensador de filtro del convertidor*)



Ilustración 80 - Condensadores EPCOS 470uF 450V. Fuente: Aliexpress

#### 4.3.15. Diseño de PCB

Para realizar esta placa de circuito impreso, se debe tener en cuenta la elevada tensión que puede haber entre la red de Vout (400V) y la de GND (0V). Esto nos limita a la hora de colocar ciertas pistas junto a otras o a garantizar una distancia mínima entre ellas.

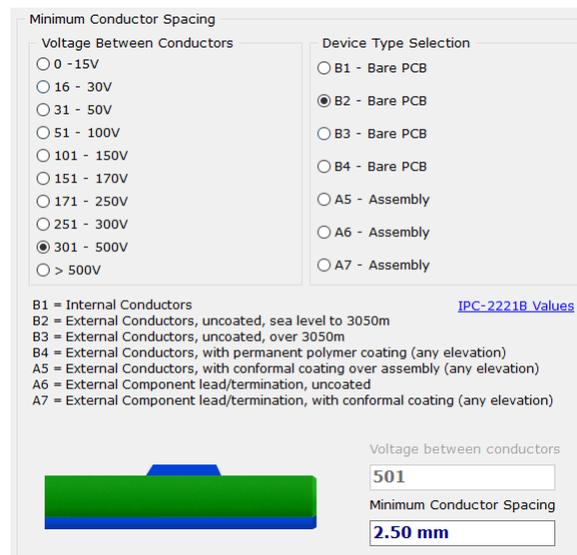
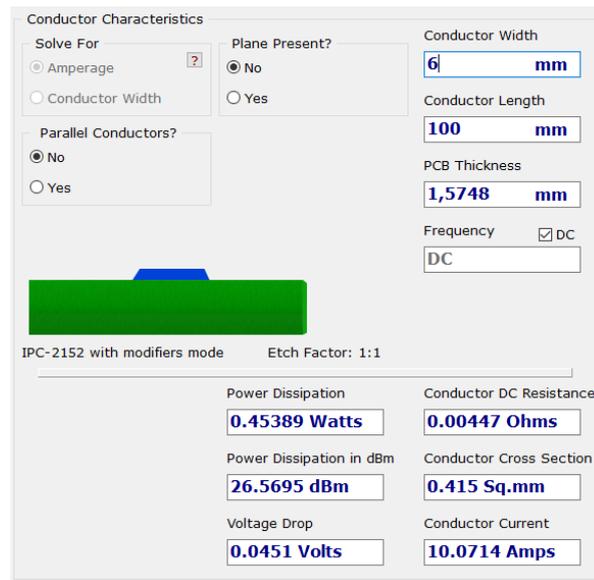


Ilustración 81 - Espacio mínimo entre pistas para tensiones de 500V. PCB Saturn Design

Podemos observar que SATURN PCB Toolkit, nos indica dejar un espacio mínimo de 2.5mm entre pistas con tensiones en el rango entre 301 y 500V hasta 3050m de altitud. En este diseño, hay un mínimo de 2.8mm de distancia entre pistas, por lo que se cumple con la especificación.

Otro parámetro a tener en cuenta, es el ancho de las pistas respecto a la corriente que circula por ellas. Esto es importante de cara a tener un incremento de temperatura adecuado.

Considerando una corriente media de 10A, mucho mayor que la que circula por el circuito, SATURN PCB Toolkit, nos recomienda un ancho de pista de al menos 6mm para un incremento de temperatura de 20°C sobre la temperatura ambiente (25°C). En nuestro diseño, el ancho mínimo en el circuito es de 5 mm, aunque es menor, podemos recubrir las pistas con estaño, lo que disminuye la resistencia de las pistas y permite circular corrientes hasta un 40% mayores para un mismo incremento de temperatura.



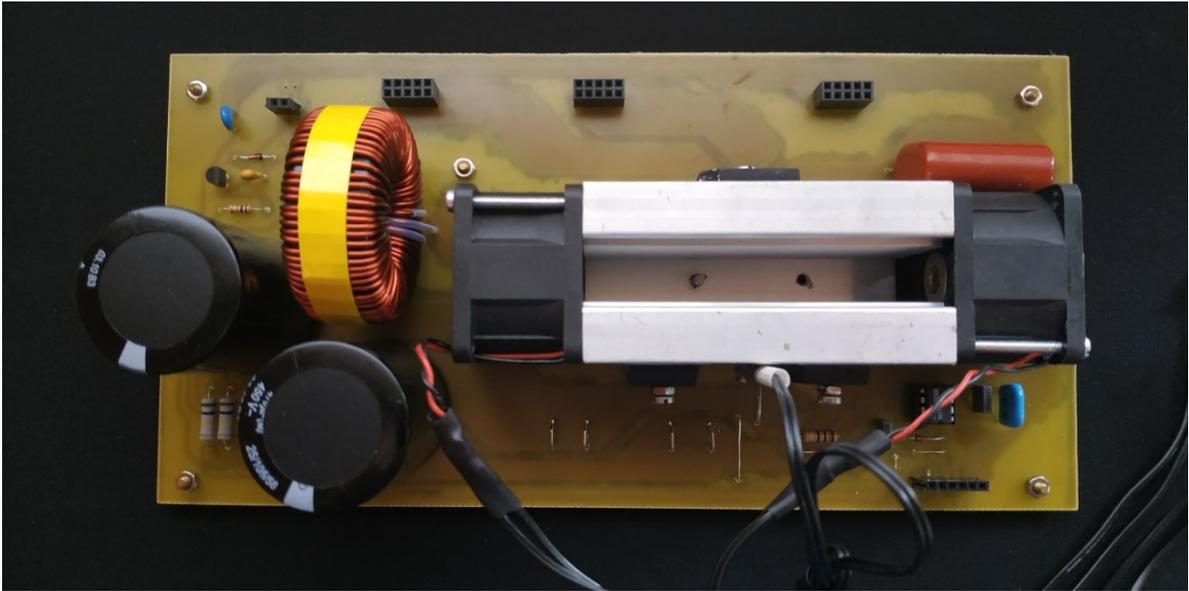
*Ilustración 82 - Corriente soportada por una pista de 6 mm de ancho y 35 um de espesor. PCB Saturn Design*

#### 4.3.16. Fabricación PCB

Como se puede observar, se han recubierto las pistas por las que puede circular gran cantidad de corriente, con estaño. En la implementación, también, dada la complejidad de la placa, se pueden ver cables que conectan zonas que se cablearon por error; estos problemas de diseño han sido corregidos en los PCB que se presentan en planos.

En la parte inferior derecha de la imagen, se puede observar que se ha dotado al soporte que une la placa con la carcasa, de conexión a tierra. Esto se hace para evitar que cuando fase o neutro se desconecten, y toquen la carcasa, ésta tenga un potencial eléctrico peligroso para las personas.

De esta manera, al tocar la carcasa fase o neutro, actuaría el diferencial de la instalación eléctrica.



*Ilustración 83 - PCB completa convertidor Boost PFC*



*Ilustración 84 - PCB completa convertidor Boost PFC. Vista inferior. Nótese que se han cubierto las pistas más importantes con estaño*

### 4.3.17. Controlador PFC

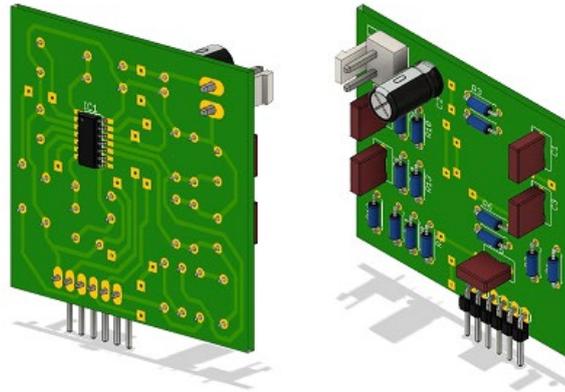


Ilustración 85 - Modelo 3D controlador PFC

Esta tarjeta de montaje vertical se encarga de medir la tensión y la corriente de pico generados por el convertidor, y ajustar el ciclo de trabajo del MOSFET en consecuencia. Se basa en un controlador del fabricante INFINEON, modelo ICE3PCS01.



Ilustración 86 - Controlador ICE3PCS01G. Fuente: Aliexpress

### 4.3.18. Configuración de la frecuencia de conmutación

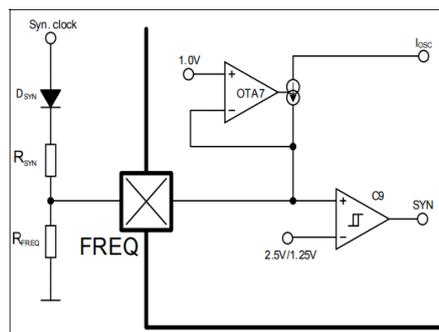


Ilustración 87 - Circuitería de selección de frecuencia. Fuente: Datasheet ICE3PCS01G

El controlador debe hacer funcionar el convertidor a 100KHz. Para ello, dispone de un pin en el que variar la frecuencia de conmutación mediante un valor óhmico determinado. Para seleccionar una frecuencia de 100KHz, debemos colocar una resistencia de 43kΩ, como se observa en esta gráfica.

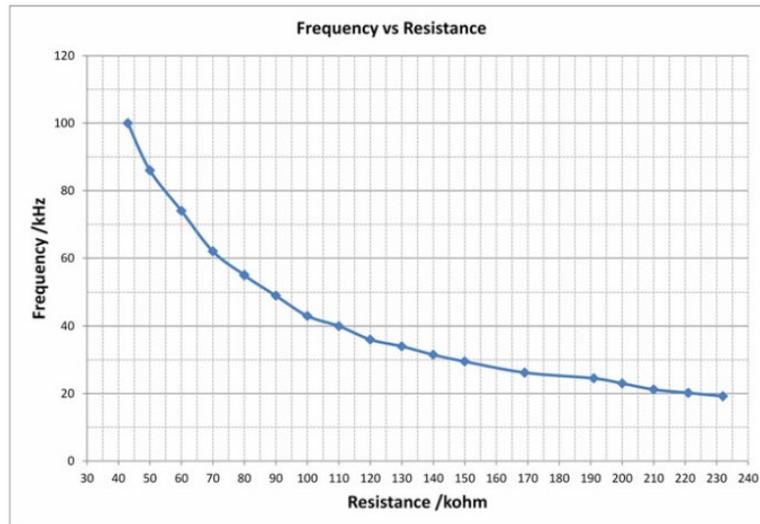


Ilustración 88 - Frecuencia según valor óhmico utilizado. Fuente: Datasheet ICE3PCS01G

#### 4.3.19. Arranque suave

Este dispositivo es compatible con arranques suaves basados en relés. Cuando la tensión de salida, supera el 95% de la tensión nominal, el pin VB\_OK se pone a nivel alto (5V). Esto se puede aprovechar implementando un circuito sencillo para manejar el relé con un transistor bipolar.

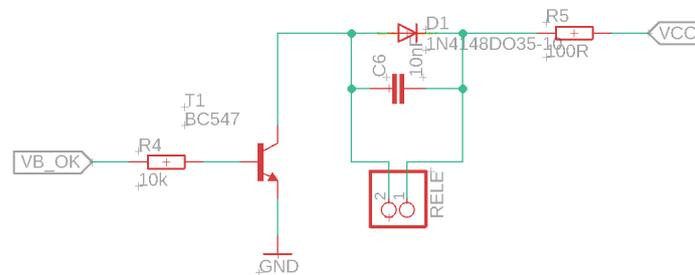


Ilustración 89 - Transistor encargado de manejar el relé de arranque suave

4.3.20. Lazo de tensión

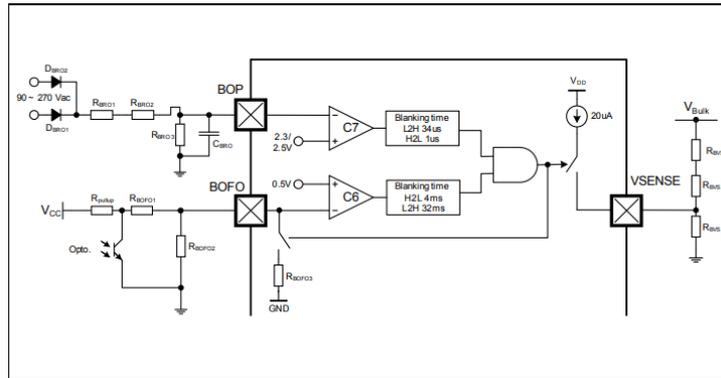


Ilustración 90 - Circuitería del lazo de tensión. Fuente: Datasheet ICE3PCS01G

Para determinar la tensión presente en la salida, el controlador dispone de varios pines de lectura de tensión y protección de sobretensión y lazo abierto.

Dado que la tensión de salida es muy alta, se debe establecer una red de resistencias, que la reduzca hasta un rango de 2.7V como máximo.

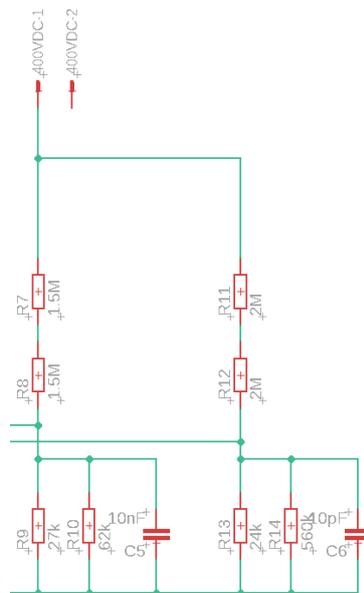


Ilustración 91 - Divisores de tensión para VSENSE y OVP

Se trata de dos divisores de tensión, uno para la lectura VSENSE, y otro para OVP. Reducen la tensión a 2.49V y 2.29V. VSENSE tiene integrada una protección de tensión llamada OVP1, que detecta cualquier nivel por encima de 2.7V y apaga el convertidor.

OVP, es la segunda protección de tensión (OVP2), que detecta cualquier tensión por encima de 2.5V y apaga el convertidor. Estas protecciones, tienen la función de evitar daños en el controlador y sobre todo, en los condensadores de salida del convertidor, que generalmente, soportarán tensiones nominales de 450V. Esto permite ahorrar en coste y espacio frente a condensadores de mayor tensión.

El controlador también integra una protección en lazo abierto OLP, que apaga el convertidor cuando la tensión en VSENSE baja de 0.5V. Se vuelve a iniciar el ciclo de arranque suave una vez sube la tensión.

#### 4.3.21. Lazo de corriente

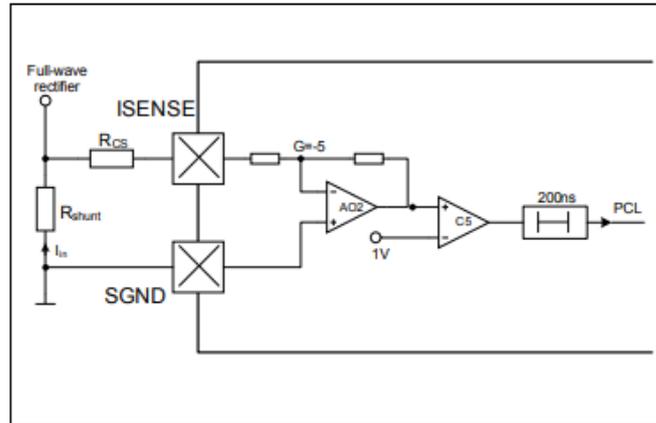


Ilustración 92 -Circuitería del lazo de corriente. Fuente: Datasheet ICE3PCS01G

El controlador dispone de una entrada para leer corriente desde un shunt en el lado de baja. La tensión en este pin, puede valer como máximo -0.2V. Después, esta tensión es amplificada por un factor de -5 y comparada con una referencia de 1V.

Para garantizar un arranque adecuado, el shunt debe dimensionarse a un valor óhmico bajo, ya que las grandes corrientes de inrush en el arranque, podrían activar esta protección.

Se ha elegido una resistencia de shunt de 0.01Ω y 5W.

#### 4.3.22. Driver MOSFET



Ilustración 93 - Driver IR2125. Fuente: Aliexpress

Para manejar el transistor, ya que el controlador no puede aportar la corriente necesaria, se ha optado por utilizar un driver del fabricante International Rectifier, modelo IR2125, capaz de aportar 2A, y trabajar a unas tensiones entre 12 y 18V, suficientes para esta aplicación. En la imagen se puede observar la configuración que se ha implementado.

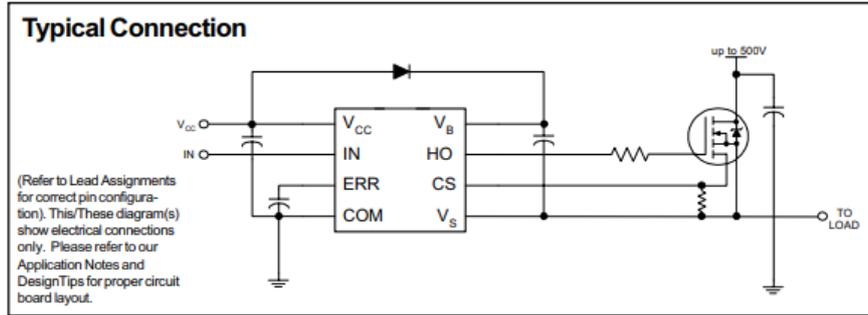


Ilustración 94 - Conexión típica driver IR2125. Fuente: Datasheet IR2125

4.3.23. Resultado PCB

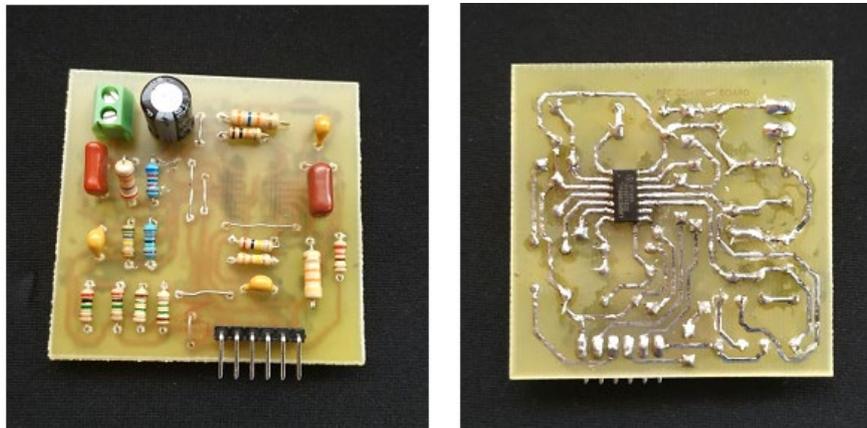


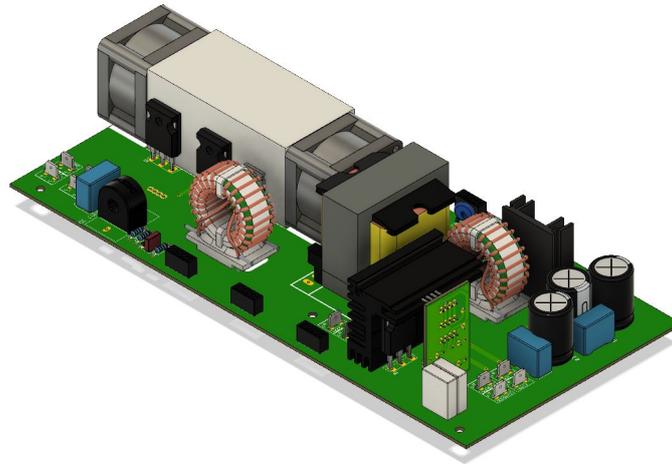
Ilustración 95 - PCB controlador PFC

4.3.24. Montaje final



Ilustración 96 - Montaje completo filtro EMI y convertidor PFC

### 4.4. Convertidor DC – DC Puente Completo ZVS



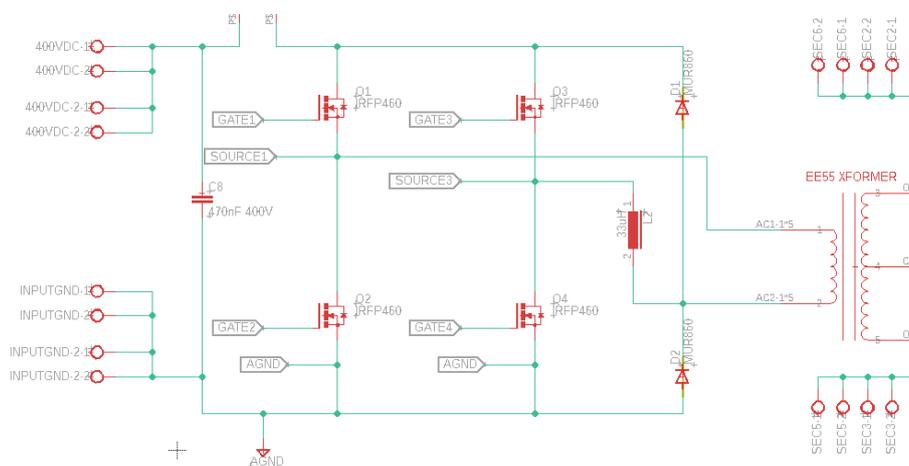
*Ilustración 97 - Modelo 3D Convertidor DC - DC ZVS*

Una vez se genera un nivel de continua constante para compensar el factor de potencia, se debe diseñar un dispositivo capaz de proporcionar aislamiento galvánico y un nivel de tensión más pequeño o más grande, según nuestro interés. En este caso, se ha optado por implementar un convertidor de puente completo, que reduce la tensión entre 0 - 60V.

Además, dado la potencia que debe entregar, se ha implementado una versión del puente completo con tanque de resonancia basado en una bobina en serie con el transformador, para alcanzar el régimen ZVS (Zero Voltage Switching) al 60% de la carga.

A continuación, se relata el dimensionamiento de las distintas partes que forman el convertidor.

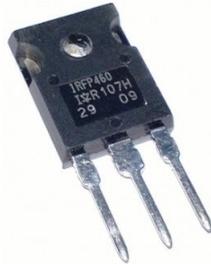
#### 4.4.1. Dimensionamiento puente completo



*Ilustración 98 - Esquema implementado del primario del convertidor DC - DC*

El principal componente que garantiza el funcionamiento del puente, son los MOSFET encargados de la conmutación. Seguidos por los diodos de vuelta que conducen en estados intermedios (Free-Wheeling). Así como la bobina de resonancia  $L_r$  y el transformador.

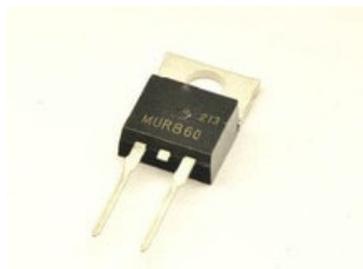
#### 4.4.2. MOSFET puente completo



*Ilustración 99 - Transistor MOSFET IRFP460. Fuente: Aliexpress*

Como conmutador principal, se ha elegido un MOSFET del fabricante VISHAY, modelo IRFP460. Es el mismo modelo que en el convertidor encargado de la corrección del factor de potencia. Se puede comprobar en el anexo de cálculos que cumple con las especificaciones de diseño. (Véase en el Anexo – Cálculos *Dimensionamiento MOSFET puente completo*)

#### 4.4.3. Diodo volante



*Ilustración 100 - Diodo MUR860. Fuente: Aliexpress*

En los estados intermedios, periodos de tiempo en los que se espera para activar el siguiente grupo de MOSFET, se produce un efecto llamado *Free-Wheeling* en el que circula corriente del transformador por el diodo integrado en los MOSFET.

Para apoyar a esta conducción, se integran unos diodos ultrarrápidos en paralelo, del fabricante FAIRCHILD, modelo MUR860, que aguantan hasta 600V en contra y pueden conducir hasta 8A.

#### 4.4.4. Diseño circuito de medida transformador de corriente

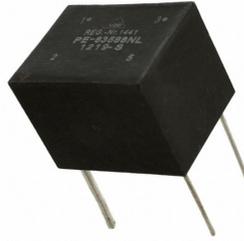


Ilustración 101 - Transformador de corriente PULSE PE-63587NL Fuente: PULSE Electronics

Para medir la corriente que circula por el transformador, se utiliza un transformador de corriente del fabricante Pulse Electronics, modelo PE-63587NL. Se debe utilizar un transformador debido que se coloca en el lado de alta tensión en el primario. Si se quisiera medir con una resistencia (Shunt), se tendría que utilizar un amplificador de aislamiento con otra alimentación adicional. De esta manera, se pierde precisión, pero es más económica y sencilla de implementar.

Conocer la tensión en el primario es muy interesante para poder predecir mejor la respuesta dinámica del sistema con el controlador. Intentar controlar el convertidor desde el secundario daría como resultado un controlador más lento, ya que se producen retrasos en la respuesta de la corriente debido al transformador, lo que hace difícil saber que va a pasar.

La medida con un transformador de corriente se basa en hacer circular la corriente principal por el primario, y leer la caída de tensión en una resistencia generada por una corriente equivalente en el secundario. En este transformador, existe una relación de 100:1 para el secundario:

$$a_2 = \frac{I_p}{I_s} = 100$$

Para calcular la corriente de pico en el primario del convertidor se utiliza:

$$I_p = \frac{1000V}{380V} \times \sqrt{2} = 3.72A$$

Se define una tensión máxima a la salida del transformador, en este caso 4V. Por lo que calculamos las resistencias encargadas de generar este valor:

$$R_{sense} = \frac{V_p - 0.2}{\frac{I_p}{a_2} \times 1.1} = \frac{4 - 0.2}{\frac{3.72}{100} \times 1.1} = 92.86 \text{ ohmios}$$

Por lo que se decide utilizar dos resistencias de 180 ohmios en paralelo para conseguir mejor precisión media. El circuito implementado es el siguiente:

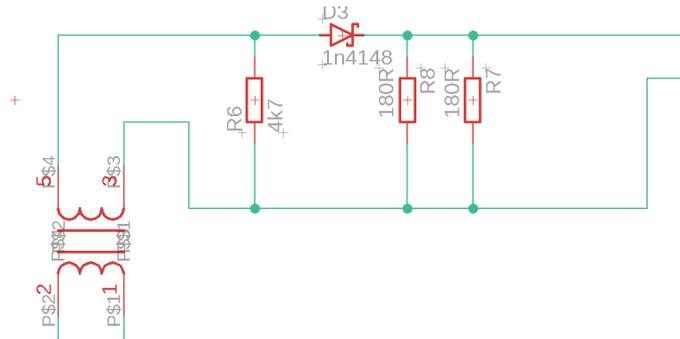


Ilustración 102 - Circuito de medida de corriente en el primario implementado

#### 4.4.5. Diseño transformador

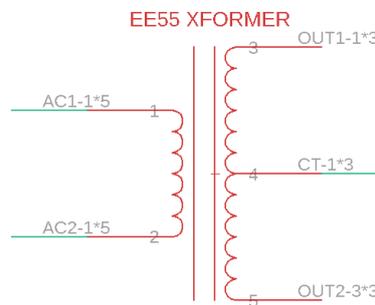


Ilustración 103 - Esquema del transformador

Una de las partes fundamentales del convertidor, es el transformador. Proporciona aislamiento galvánico, a la vez que transmite la energía de un lado a otro.

Dada la frecuencia de funcionamiento de las fuentes conmutadas, los núcleos magnéticos de los transformadores no pueden ser de Acero-Silicio ni en forma de láminas. Esto provocaría enormes pérdidas de saturación en el núcleo, por lo que se construyen a partir de materiales ferromagnéticos llamados generalmente “ferrita”. El más utilizado por su relación calidad – precio, es el PC40.

Se han realizado los cálculos para una tensión de entrada de 400V, una tensión a la salida de más de 60V para tener margen de regulación y 100 KHz de frecuencia de operación. De esta manera, se han bobinado:

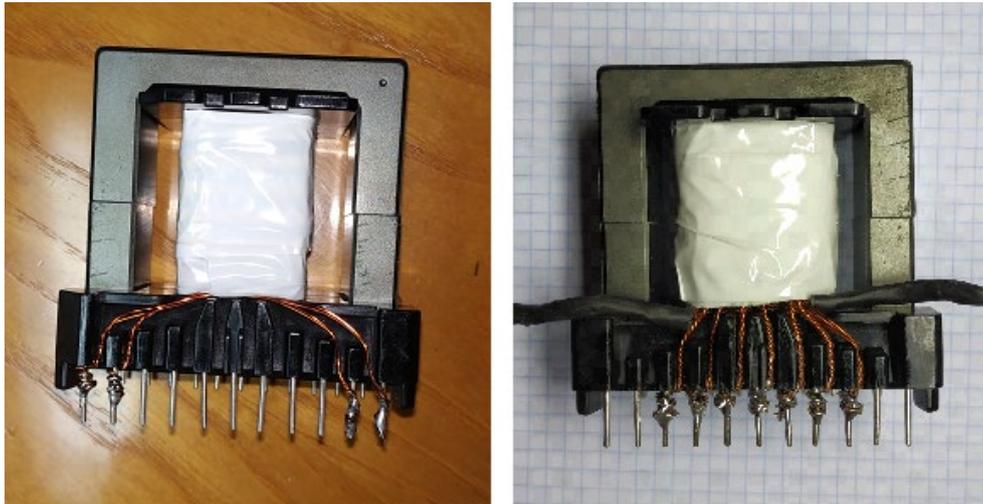
	Nº de vueltas	Nº de cables	Capacidad (A)
Primario	18	4	10.8
Secundario	4 + 4	14	37.8

Los cálculos del dimensionamiento, se encuentran se en el Anexo – *Cálculos Diseño Transformador de la fuente conmutada.*

Se han utilizado un total de 28 metros de cable para construirlo.

Se ha elegido como transformador, un núcleo en formato EE55/28/21, de material PC40, dada su disponibilidad en bajas cantidades y bajo precio. De hecho, como se puede observar en los cálculos, este núcleo está muy sobredimensionado, pero para acceder

a núcleos más pequeños y eficientes como el PQ4040, se debe realizar un pedido mínimo (lo que dificulta la implementación de prototipos).

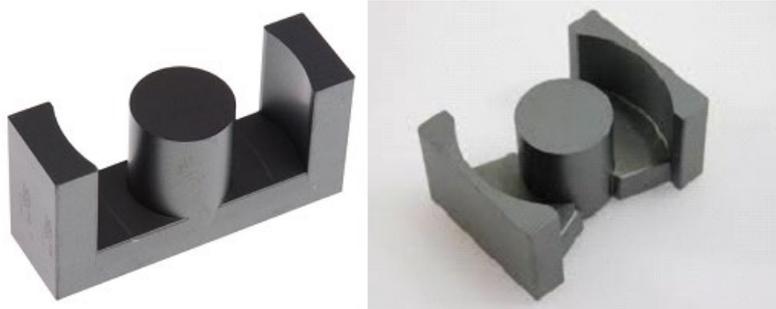


*Ilustración 104 - Izquierda: transformador con el bobinado primario. Derecha: transformador completado*

#### 4.4.6. Alternativas transformador

Dado que el transformador tiene todavía bastante hueco en la ventana, se podrían hacer los cálculos para una magnetización menor y así tener que usar más cobre para llegar a los requerimientos. De esta manera se mejoraría el comportamiento magnético del transformador.

Otra alternativa, sería utilizar, dado que el EE55/28/21 está muy sobredimensionado en área de núcleo para esta aplicación, un núcleo más pequeño. Entre los candidatos, destacan el ETD49 y el PQ4040, que, por la forma cilíndrica de la columna central, permiten apilar mejor los bobinados y son más eficientes.



*Ilustración 105 - Izquierda (Núcleo ETD49) Derecha (Núcleo PQ4040)*

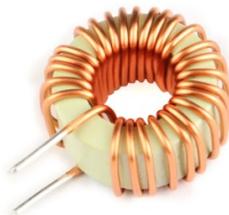
#### 4.4.7. Terminales Faston 6.3mm



*Ilustración 106 - Terminales Faston hembra (Izquierda) y macho (derecha). Fuente: Aliexpress*

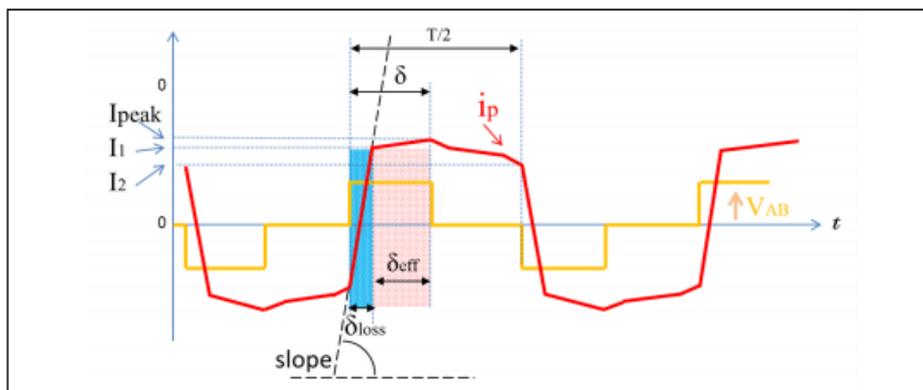
Para conectar el transformador a la placa de circuito impreso, no se ha podido realizar todo el cableado en forma de pistas por falta de espacio, por lo que se ha optado por conectar la toma de centro del secundario a la placa, y los otros dos extremos terminarlos en conectores Faston de 6.3mm y 25A. Esto nos permite colocarlos en otra parte del circuito más cómodamente.

#### 4.4.8. Bobina de resonancia



*Ilustración 107 - Bobina 33uH y 10A. Fuente: Aliexpress*

Para poder alcanzar la condición de ZVS, debemos modificar la pendiente con la que circula la corriente en el transformador. Según cómo dimensionemos la bobina, haremos la pendiente más o menos grande. Generalmente, se suele dimensionar para alcanzar régimen ZVS entre el 50 y el 100% de la carga total del convertidor.



*Ilustración 108 - Pendiente deseada en la corriente del convertidor. Fuente: STEVAL-ISA172V2: 2 kW fully digital AC - DC power supply. ST Microelectronics*

En el apartado de Anexo – Cálculos *Dimensionamiento Bobina de resonancia*  $L_R$ , se obtiene que para cumplir la condición de ZVS al 57% de la carga aproximadamente, se requiere una bobina de 33uH, por lo que se ha elegido una de este valor y capacidad para soportar 10A.

#### 4.4.9. Red RCD contra picos de tensión y oscilaciones

Por efecto de las conmutaciones en el primario, se producen picos de tensión en los MOSFET de rectificación síncrona en el secundario. Es por ello, que se recomienda colocar una red RCD en el secundario para atenuar estos efectos.

Se ha extraído este circuito de una red implementada en el documento “Phase-Shifted Full Bridge DC/DC Power Converter Design Guide” de Texas Instruments.

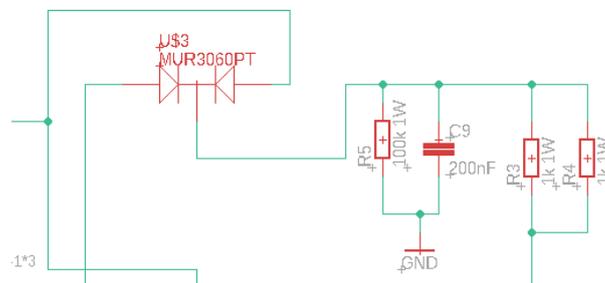


Ilustración 109 - Snubber RCD implementado

Para hacernos una idea de la tensión que deben soportar dichos transistores:

$$V_{secundario} = 2 \times \frac{N_s}{N_p} \times V_{in} = 2 \times \frac{4}{18} \times 400V = 178V$$

Por lo que los MOSFET elegidos, deberán aguantar por lo menos 200V entre fuente y drenador.

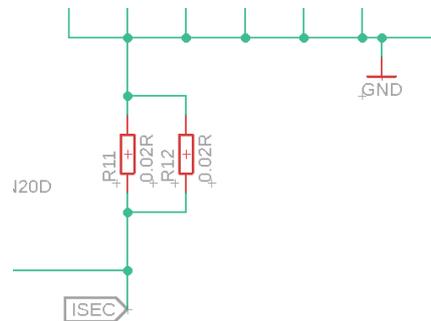
#### 4.4.10. Resistencia de medida de corriente (Shunt)



Ilustración 110 - Shunt 0.02 ohmios utilizado. Fuente: Aliexpress

Para medir la corriente que circula en el secundario, se ha elegido colocar una resistencia de medida (Shunt) en el lado de baja tensión, de forma que genere una caída

de tensión detrás de la referencia GND, es decir, dicha tensión será negativa. Se puede apreciar mejor en la siguiente imagen:



*Ilustración 111 - Circuito de medida de corriente en el secundario*

El shunt elegido, está formado por dos resistencias de 0.02 ohmios y 5W, colocadas en paralelo, formando un valor de 0.01ohmios. Dichos valores responden a los siguientes cálculos:

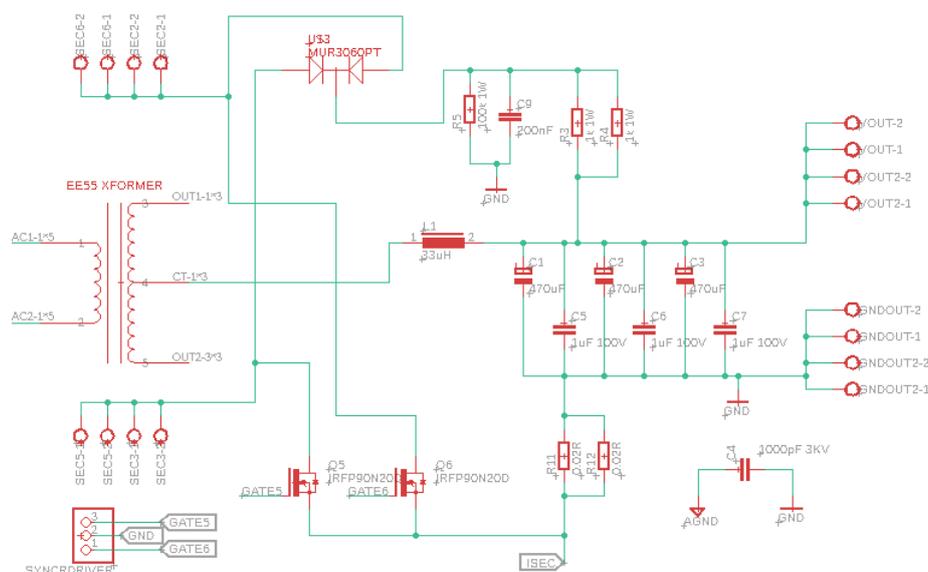
$$\text{Caída Shunt} = I_{\text{secundario}} \times R_{\text{shunt}} = 16A \times 0.01\Omega = 160 \text{ mV}$$

Lo que nos da una potencia disipada en el shunt:

$$\text{Pérdidas Shunt} = R_{\text{shunt}} \times I_{\text{secundario}}^2 = 0.01\Omega \times 16^2 = 2.56 \text{ W}$$

#### 4.4.11. Rectificación síncrona en el secundario

Se ha decidido implementar una rectificación síncrona en el secundario, ya que como se ha explicado anteriormente, se reducen considerablemente las pérdidas frente al uso de diodos convencionales.



*Ilustración 112 - Secundario del convertidor DC - DC*

#### 4.4.12. MOSFET rectificación síncrona



Ilustración 113 - Transistor MOSFET IRFP90N20D

El MOSFET elegido para esta tarea, es del fabricante International Rectifier modelo IRFP90N20D, cuyas características principales son una  $R_{ds(ON)}$  de tan solo 23 miliohmios y una tensión drenador – fuente de 200V, por lo que puede soportar los picos de tensión en el secundario. Se puede observar la mejora en las pérdidas en el apartado de Anexo – Cálculos *Dimensionamiento MOSFET rectificación síncrona*.

#### 4.4.13. Rectificación síncrona Autónoma (Self Driven)

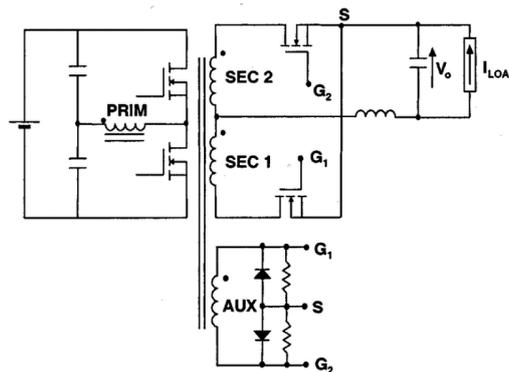


Ilustración 114 - Esquema básico rectificación síncrona autónoma. Fuente: *The Implication of Synchronous Rectifiers to the Design of Isolated, Single-Ended Forward Converters*. Texas Instruments.

Para facilitar el control del secundario, se ha decidido implementar una rectificación autónoma, de la que no nos tengamos que preocupar, en la que se activen automáticamente los transistores con la tensión del transformador. Esto tiene el inconveniente de ser menos eficiente, ya que entran en conducción los diodos integrados en el cuerpo de los MOSFET en los intervalos en los que no están activos.

Dada la relación de transformación en el transformador, podemos estimar que, con un bobinado de una vuelta para cada transistor con una toma de centro, obtenemos 20V en la puerta de los MOSFET.

#### 4.4.14. Diseño etapa de filtrado en el secundario

Para el filtrado de la corriente rectificada en el secundario, se ha utilizado una estructura de una bobina en serie con la carga y condensadores en paralelo.

En base al dimensionamiento requerido, (Véase en el Anexo – Cálculos *Diseño etapa de filtrado en el secundario*) se coloca una bobina de 33uH y 28A.

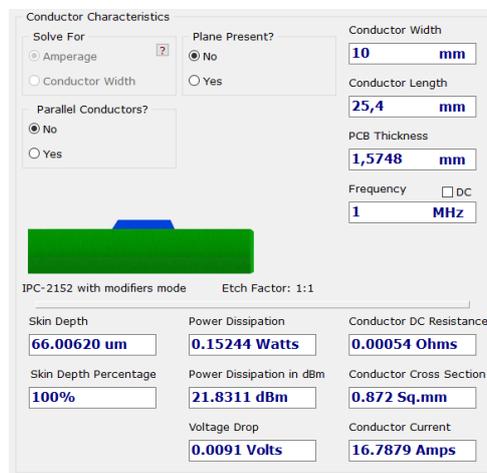


*Ilustración 115 - Condensadores Nichicon 470 uF 100V. Fuente: Aliexpress*

Para cumplir ampliamente la capacidad necesaria calculada en el anexo, se colocan tres condensadores de 470uF en paralelo. Se han utilizado condensadores electrolíticos del fabricante Nichicon, de baja ESR. Se añaden tres condensadores de polyester de 1uF y 250V para mejorar la respuesta transitoria de los electrolíticos. (Véase en el Anexo – Cálculos *Diseño etapa de filtrado en el secundario*)

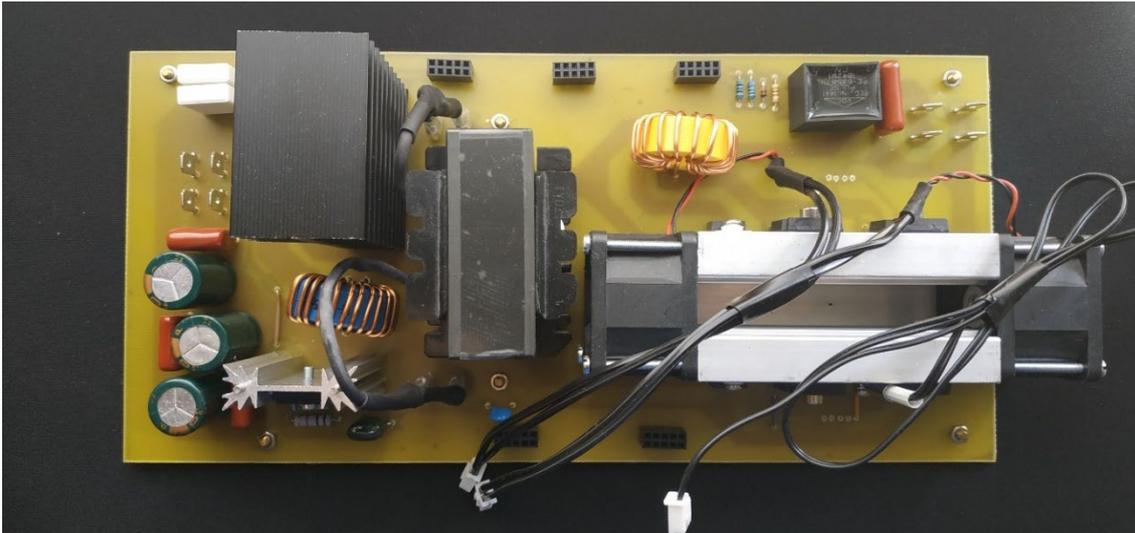
#### 4.4.15. Construcción

En el diseño del circuito impreso de este convertidor se ha vuelto a respetar la distancia de 2.5 mm entre pistas en el primario dada la gran diferencia de tensión entre ellas. Además, se ha dejado un mínimo de 8 mm de separación entre pistas del primario y del secundario para garantizar aislamiento galvánico entre ellas.

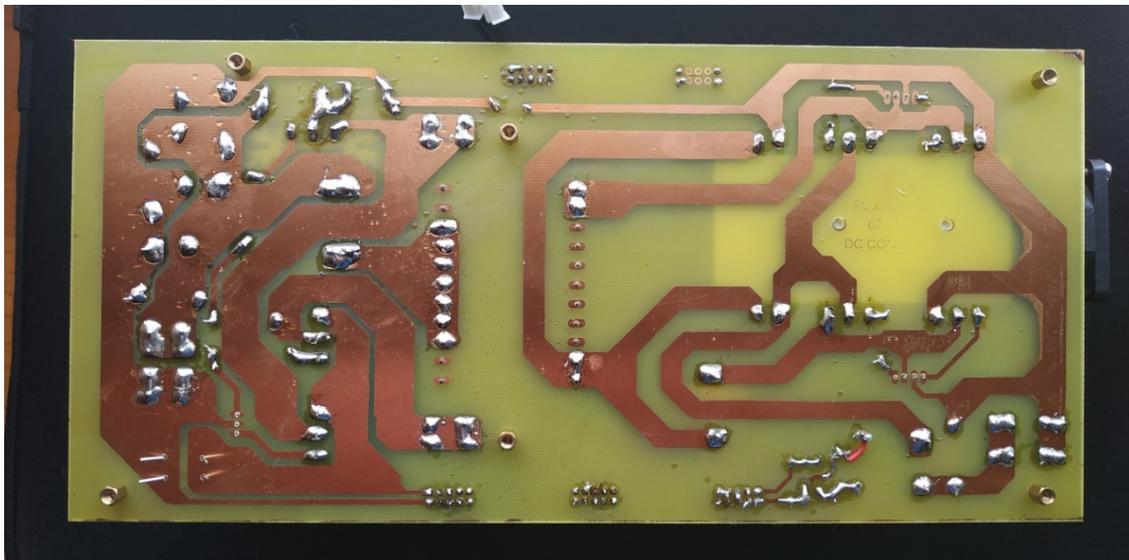


*Ilustración 116 - Capacidad de corriente para pistas de 10 mm y 35 um de espesor. PCB Saturn Design*

Se han dimensionado las pistas del secundario para poder soportar 16A de intensidad. Para esto se necesita un ancho de pista de 10 mm, para un incremento de temperatura de 20°C, sobre la temperatura ambiente (25°C), pero por cuestiones de espacio, hay un tramo que tiene 7 mm. Esto no es problema ya que posteriormente se recubrirá con estaño.



*Ilustración 117 - PCB Completo convertidor DC - DC. Vista superior*

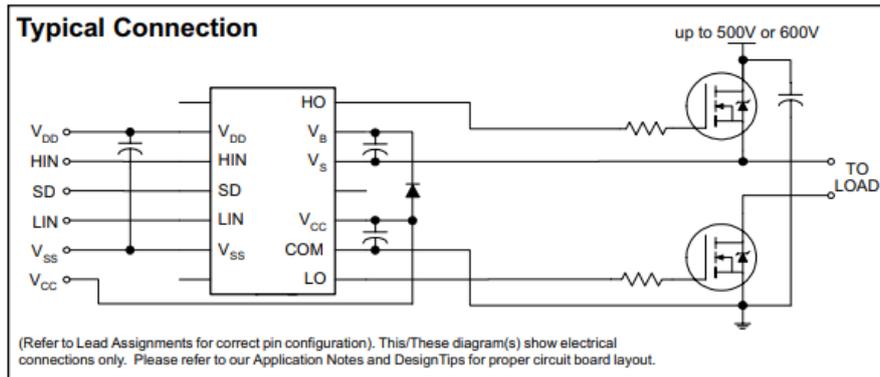


*Ilustración 118 - PCB completo convertidor DC - DC. Vista inferior*

## 4.5. Drivers MOSFET Puente Completo (Primario)

### 4.5.1. Drivers en el lado de alta tensión vs lado de baja tensión

Las fuentes de alimentación, tradicionalmente, incluyen el driver y su controlador dedicado en el lado de alta tensión a los que llega alimentación a través de un bobinado auxiliar en el transformador. El diagrama de conexión de dichos drivers tiene el siguiente aspecto:



*Ilustración 119 - Conexión típica driver IR2110 en el lado de alta tensión. Fuente: Datasheet IR2110*

Para nuestro diseño, se podría haber dejado tanto los drivers, como el microprocesador del controlador principal en el lado de alta tensión y comunicarnos con el procesador de la interfaz con el PC con un optoacoplador.

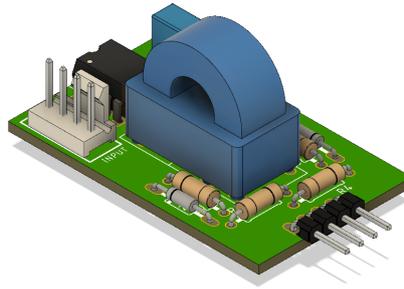
De la misma forma, se podrían haber dejado los drivers en el lado de alta tensión y el microcontrolador en el lado de baja, mandando las señales de conmutación, de nuevo, con un optoacoplador.

La gran desventaja que presenta esta estructura, es que necesitamos dos fuentes de alimentación distintas (deben estar aisladas galvánicamente entre ellas) y eso resulta en ocupar espacio y un mayor coste.

La solución más extendida en las notas de aplicación de los fabricantes de estos sistemas, pasa por colocar los drivers y el procesador en el lado de baja tensión, y conmutar los transistores a través de transformadores de pulsos integrados en los propios drives.

De esta manera, se elimina la necesidad de otra fuente de alimentación adicional y se garantiza el aislamiento galvánico entre el convertidor y su sistema de control, ya que, de otro modo, gobernar un circuito por el que circulan 400V, con otro de 12V, puede tener grandes consecuencias en caso de avería.

### 4.5.2. Drivers desarrollados



*Ilustración 120 - Modelo 3D del driver puente completo con aislamiento para el lado de alta tensión*

Para gobernar los transistores MOSFET que forman el puente completo encargado de controlar la tensión en el transformador desde el lado primario, se utilizan dos drivers. Uno por cada dos transistores, habiendo cuatro en total.

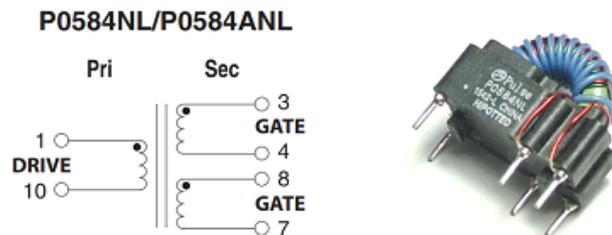
Dado que el microcontrolador principal, no es capaz de proporcionar corriente suficiente para manejar las puertas de un transistor MOSFET, se utiliza un circuito intermedio que, mediante una línea de alimentación adicional, amplifica las señales lógicas del controlador de 5V a unas señales de 12V y varios amperios de intensidad.

El circuito integrado utilizado para esta tarea es el driver MCP1404 proporcionado por Microchip, que es capaz de manejar corrientes de hasta 4,5A. Dispone de dos entradas PWM que se ven reflejadas en la salida con el mismo ciclo de trabajo, pero a mayor tensión.



*Ilustración 121 - Driver MCP1404. Fuente: Microchip*

Se ha utilizado un transformador de pulsos de estructura 1:1:1, es decir, un primario y dos secundarios. El transformador utilizado procede del fabricante Pulse Electronics modelo P584NL. Estos dispositivos proporcionan 3000 Vrms de aislamiento entre el primario y el secundario, y hasta 1500 Vrms entre los dos secundarios.



*Ilustración 122 - Transformador de pulsos P0584NL. Esquema (Izquierda) Aspecto físico (Derecha). Fuente: Datasheet P0584NL*



Ilustración 123 - PCB Completa driver puente completo P0584NL

Para comprobar el correcto funcionamiento de estos drivers, se ha alimentado el circuito con 12VDC y se han insertado dos señales PWM al 50% de ciclo de trabajo complementarias entre sí. De tal forma que se simulen las señales reales del puente completo.

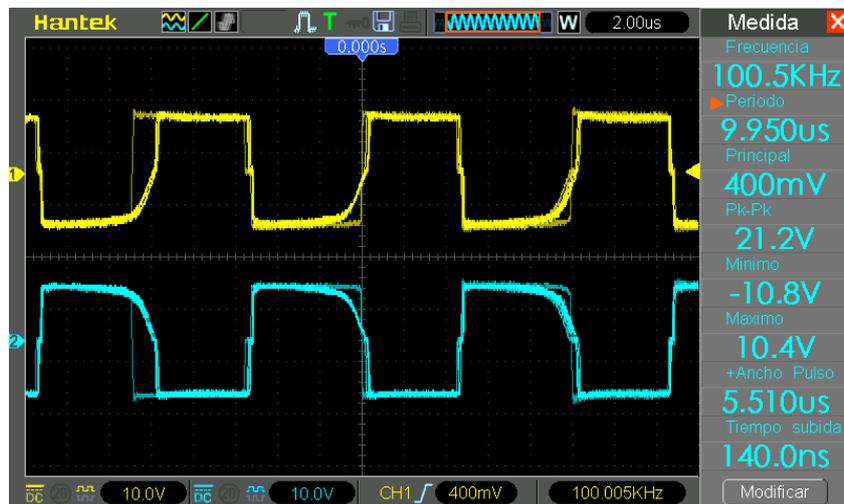


Ilustración 124 - Funcionamiento del driver de puente completo con dos señales simétricas al 50%

### 4.6. Driver MOSFET Rectificación Síncrona (Secundario)

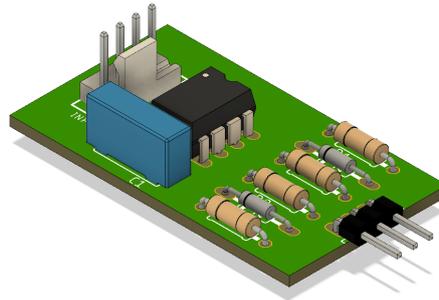


Ilustración 125 - Modelo 3D driver rectificación síncrona

Para controlar los transistores MOSFET encargados de la rectificación síncrona del convertidor DC-DC, se requiere un circuito mucho más sencillo. Dado que estos transistores se encuentran en el lado del secundario y comparten la misma referencia GND con el circuito de control, no se necesita aislamiento galvánico.

De esta manera, se conecta directamente el driver a los transistores.

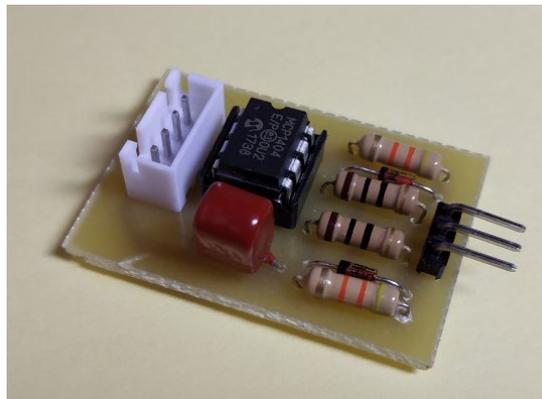


Ilustración 126 - PCB completo del driver de rectificación síncrona

Para probar el circuito se ha conectado a 12VDC y se han inyectado las dos señales complementarias anteriores. En la salida se pueden ver las señales con un ciclo de trabajo idéntico al de la entrada, pero con un nivel de 12VDC.

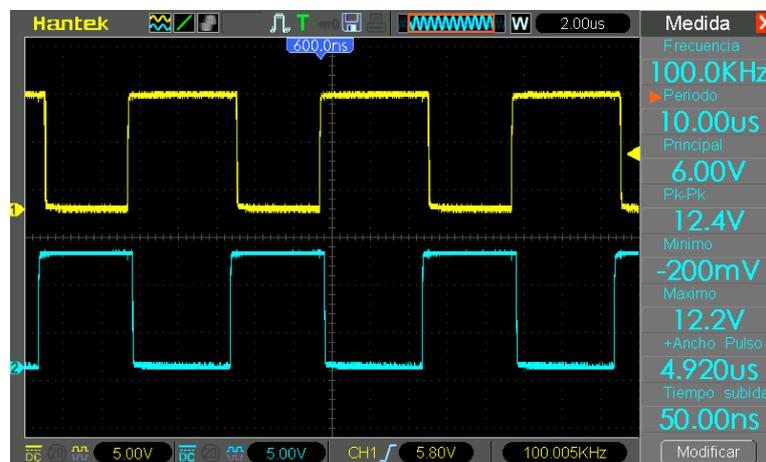


Ilustración 127 - Funcionamiento driver de rectificación síncrona con dos señales simétricas al 50%

## 4.7. Regulación de temperatura y control de ventiladores

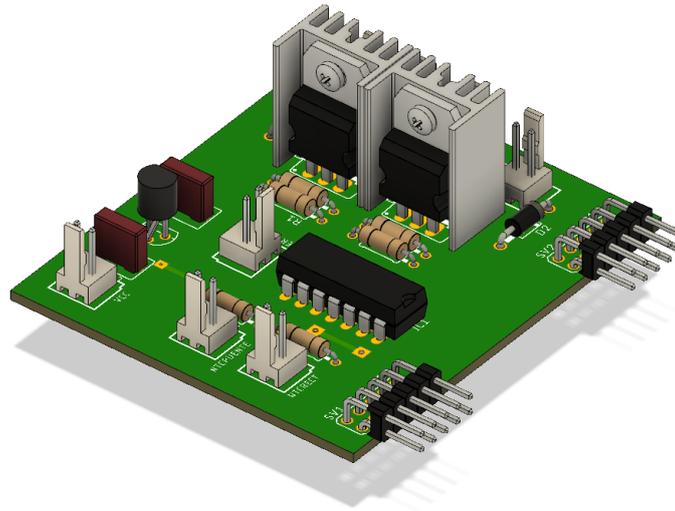


Ilustración 128 - Modelo 3D controlador de temperatura

Para desarrollar el control de temperatura de la fuente, se ha optado por un sistema de ventilación forzada basada en unos perfiles de aluminio en los que se han colocado un ventilador de 40mm en cada extremo.

Este sistema se encarga de refrigerar el rectificador, el transistor MOSFET y el diodo del convertidor PFC, así como los MOSFET del puente completo en el convertidor DCDC principal.

Se basa en la lectura de una resistencia NTC de 10 K $\Omega$ , modelo B3950, colocada en cada disipador mediante un microcontrolador ATtiny84. Éste, lee el valor de un divisor de tensión que varía según la temperatura y activa los ventiladores a una frecuencia de 32KHz (fuera del espectro audible) con un valor PWM que depende del tramo de temperatura en el que se encuentre.

### 4.7.1. Disipadores utilizados



Ilustración 129 - Disipador de aluminio 40x40x80 mm. Fuente: Aliexpress

Se han utilizado dos perfiles de extrusión de aluminio en forma de U (o túnel) para canalizar mejor el flujo de aire a través de éstos. Tienen aletas en las paredes interiores y agujeros de 3 mm sin roscar para la instalación de ventiladores de 40 mm.

Las dimensiones de cada perfil son de 40x40 mm de sección y 80 mm de largo. Tienen una capacidad de disipación térmica de 4°C/W, que con el flujo de aire de los

ventiladores a 5 m/s, podría alcanzar los  $3.32^{\circ}\text{C}/\text{W}$  aproximadamente. Se estima en los cálculos, una temperatura máxima de operación del convertidor PFC de  $89.7^{\circ}\text{C}$  y del convertidor DC – DC de  $74.8^{\circ}\text{C}$ . (Véase en el Anexo – Cálculos *Dimensionamiento del sistema de refrigeración de la fuente*)



*Ilustración 130 - Disipador CR101-50. Fuente: Mouser*

Para los MOSFET de rectificación síncrona se ha usado otro disipador sin refrigeración activa del fabricante OHMITE, modelo CR101-50, con un coeficiente térmico de  $5.3^{\circ}\text{C}/\text{W}$ . Se observa en los cálculos, una temperatura de operación de los transistores de rectificación síncrona de  $90^{\circ}\text{C}$ . (Véase en el Anexo – Cálculos *Dimensionamiento del sistema de refrigeración de la fuente*)

#### 4.7.2. Sensores utilizados y acondicionamiento

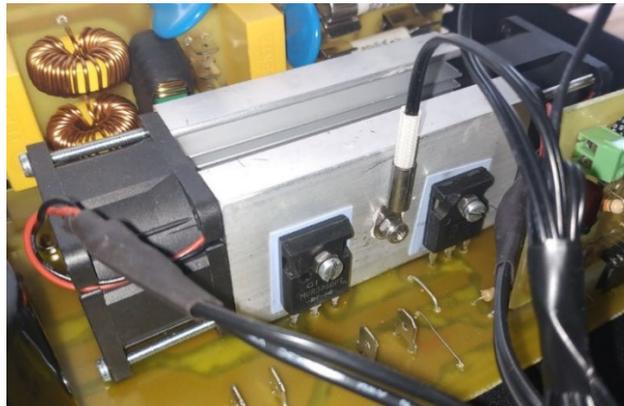


*Ilustración 131 - NTC 10K B3950. Fuente: Sparkfun*

Se trata de resistencias NTC B3950 de  $10\text{ K}\Omega$  con una precisión del 1% que funcionan en un rango de temperaturas de  $-55^{\circ}\text{C}$  a  $125^{\circ}\text{C}$ .

Incluyen un conector JST de 2 pines y un cabezal atornillable para su montaje.

### 4.7.3. Localización de los sensores



*Ilustración 132 - Vista de montaje del termistor NTC*

Se han atornillado los sensores entre dos dispositivos, en ambos disipadores, para realizar una medida más exacta de la temperatura de ambos en el disipador. Aun así, se han considerado temperaturas de lectura más bajas para variar el ciclo de trabajo de los ventiladores en el programa, por la diferencia de temperatura del disipador frente a los dispositivos.

### 4.7.4. MOSFET de control de ventiladores



*Ilustración 133 - Transistor MOSFET IRFZ44. Fuente: Aliexpress*

El manejo de los ventiladores no se puede realizar directamente con la corriente que genera el microcontrolador, por lo que se coloca una etapa intermedia formada por un transistor MOSFET.

Los transistores elegidos son los IRFZ44. Características principales:

$V_{DS}$ (V)	60	
$R_{DS(on)}$ ( $\Omega$ )	$V_{GS} = 10\text{ V}$	0.028
$Q_g$ (Max.) (nC)	67	
$Q_{gs}$ (nC)	18	
$Q_{gd}$ (nC)	25	
Configuration	Single	

*Ilustración 134 - Características generales IRFZ44*

Se coloca uno por cada dos ventiladores, es decir, uno para la etapa PFC y otro para el convertidor DC-DC principal.

De esta forma mantenemos un tamaño de la placa reducido y unas pérdidas contenidas, dado la baja  $R_{ds}$  de estos transistores. (Véase en el Anexo – Cálculos *Dimensionamiento MOSFET de control de ventiladores*)

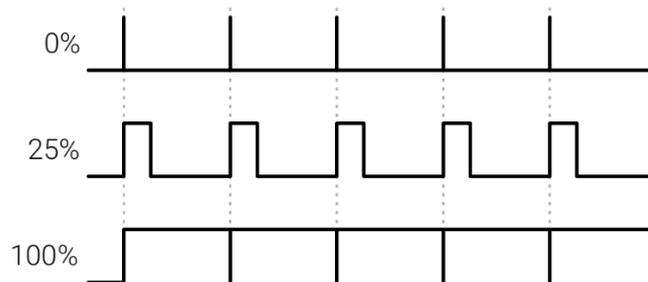
#### 4.7.5. Ventiladores utilizados



*Ilustración 135 - Ventilador Delta FFB0412SVHN. Fuente: Aliexpress*

Se han utilizado ventiladores de 40mm del fabricante Delta modelo FFB0412SVHN. Funcionan con 12VDC y se han elegido por la baja corriente que demandan, de tan sólo 0.25A. Dado que se han instalado 4 de ellos en el sistema, se genera un consumo total de 12W. De esta manera, se logra interferir lo menos posible en la eficiencia global de la fuente de alimentación.

#### 4.7.6. Control de ventiladores mediante PWM



*Ilustración 136 - Señales PWM en varios ciclos de trabajo*

Para regular la temperatura de la fuente y a su vez, eliminar el ruido innecesario de los ventiladores a bajas potencias, se ha realizado un control de velocidad PWM de éstos.

La técnica consiste en mandar pulsos a una frecuencia fija e ir variando su ancho entre el 0 y el 100%. De esta manera, cuanto más ancho sea el pulso, más tensión efectiva llegará al ventilador, aumentando su velocidad.

El programa implementado, genera una señal cuadrada con una frecuencia de 32KHz, especialmente interesante, ya que por debajo de 20KHz los motores de los ventiladores amplifican los pulsos haciéndolos audibles, resultando en un tono agudo molesto para el ser humano. De esta manera, se llevan las señales de control fuera del espectro audible, minimizando el ruido proveniente de los ventiladores.

### 4.7.7. Microcontrolador ATtiny 84

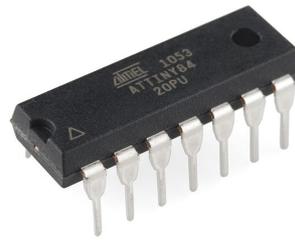


Ilustración 137 - Microcontrolador ATtiny 84. Fuente: Microchip

Para una aplicación sencilla como esta, en la que no se requieren grandes cantidades de entradas y salidas ni una gran potencia de cálculo, Se ha optado por un microcontrolador de la familia ATTINY de Atmel. Son unos microcontroladores sencillos, de patillaje reducido pero que incluyen funciones básicas, entre las que destacan un ADC de 10bits, comunicación I2C y SPI, así como varias salidas PWM.

Para nuestra aplicación se han necesitado dos entradas del ADC y dos salidas PWM, así como una salida de aviso al controlador central en caso de sobrettemperatura.

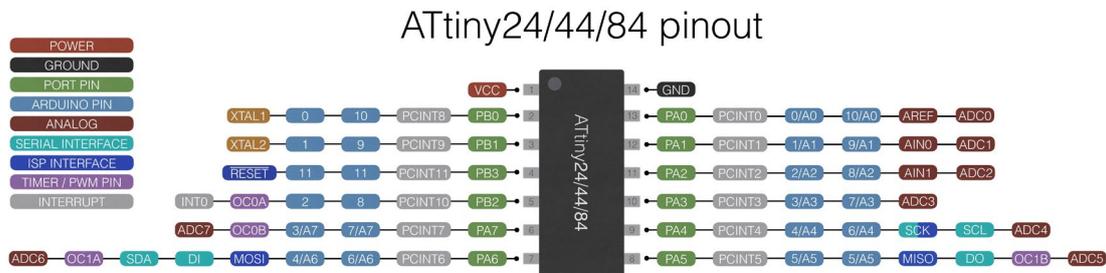


Ilustración 138 - Esquema de funciones por patillaje ATtiny84. Fuente: [homemadehardware.com](http://homemadehardware.com)

### 4.7.8. Configuración y generación de señales PWM

Para generar las señales PWM, se deben modificar los registros del Timer 1, ya que el Timer 0 es usado, generalmente, por las funciones de la IDE de Arduino. De esta manera se mantiene la compatibilidad con el entorno.

En el timer, configuramos los diferentes registros de escala del reloj para generar dos señales a 32 KHz.

### 4.7.9. Estructura del programa

El programa que se ha realizado en este controlador, consiste en la lectura de los sensores NTC mediante el convertor A/D integrado y su conversión a temperatura.

Una vez obtenida la temperatura, se varía el ciclo de trabajo de los ventiladores según el rango en el que nos encontremos. Pueden observarse en la siguiente tabla:

Rango de temperaturas	Ciclo de trabajo ventiladores
< 35 °C	0 %
35 – 40 °C	25 %
40 – 45 °C	50 %
50 – 55 °C	75 %
55 – 65 °C	100 %

A partir de 65°C se activa una salida del microcontrolador que está conectada al controlador principal y coloca la fuente de alimentación en modo “Alarma de temperatura” que mantiene los ventiladores al 100% y desactiva la salida de la fuente. . (Véase el código completo en el Anexo – Código *Programa Control de temperatura*)

Diagrama de flujo

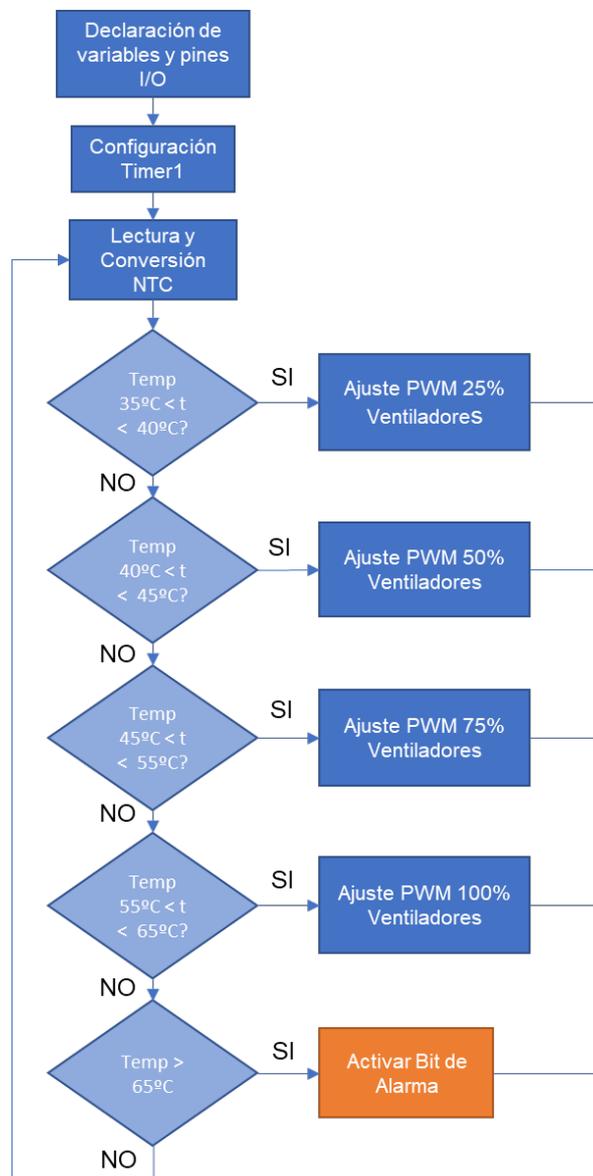


Ilustración 139 - Diagrama de flujo del programa para el controlador de temperatura

4.7.10. Circuito impreso

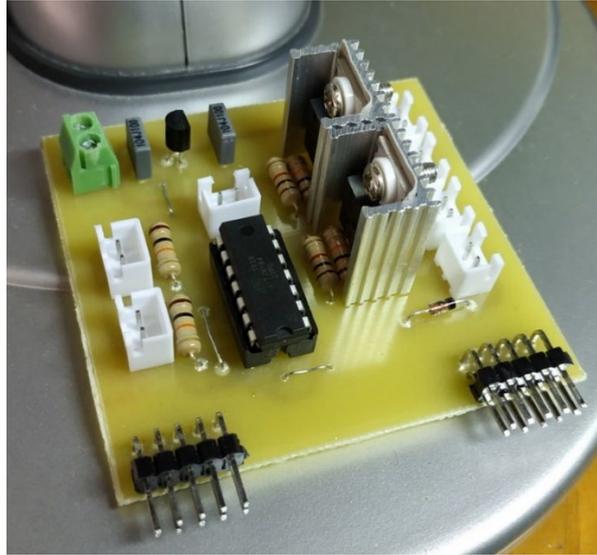


Ilustración 140 - PCB completa del controlador de temperatura

4.7.11. Funcionamiento

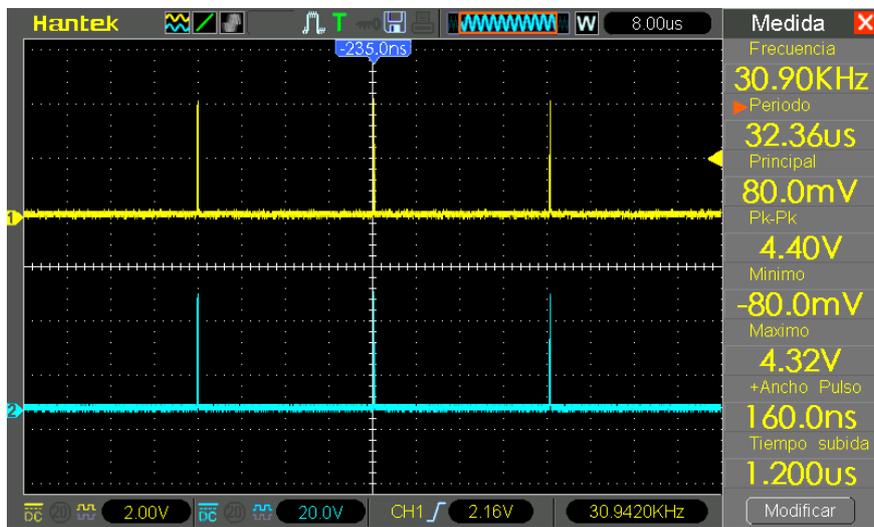


Ilustración 141 - Señales PWM al 0% con una temperatura menor de 35°C

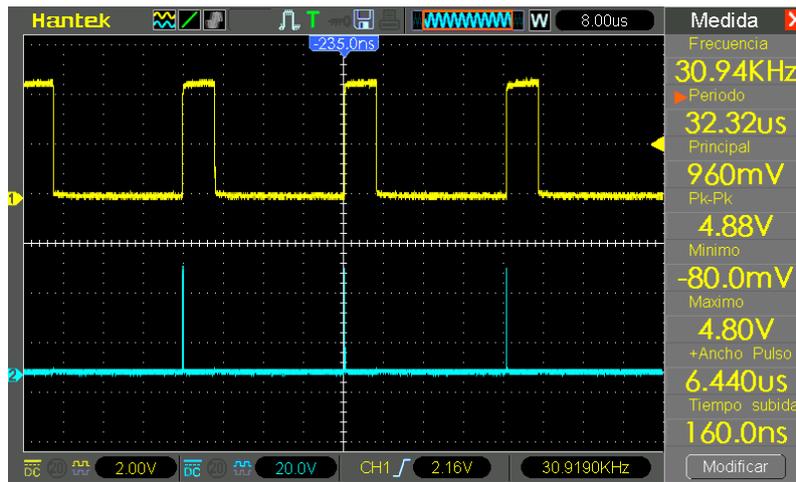


Ilustración 142 - Señales PWM al 25% para una temperatura entre 35 y 40°C

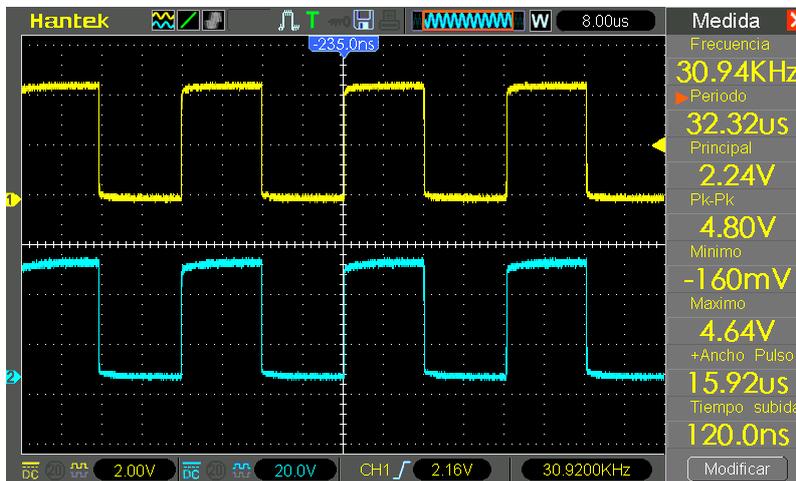


Ilustración 143 - Ilustración 136 - Señales PWM al 50% para una temperatura entre 40 y 45°C

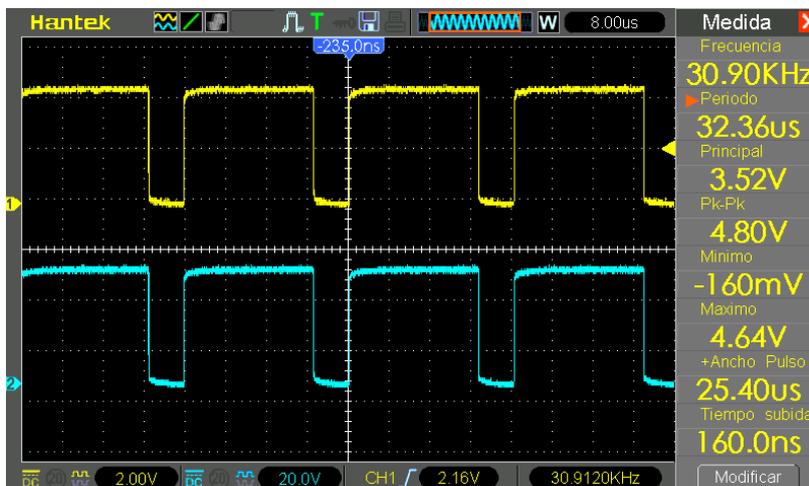


Ilustración 144 - Ilustración 136 - Señales PWM al 75% para una temperatura entre 45 y 55°C

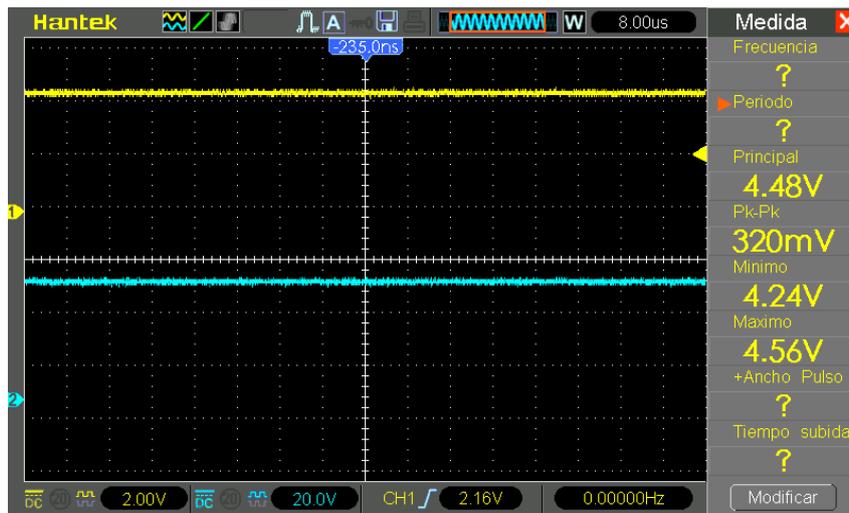


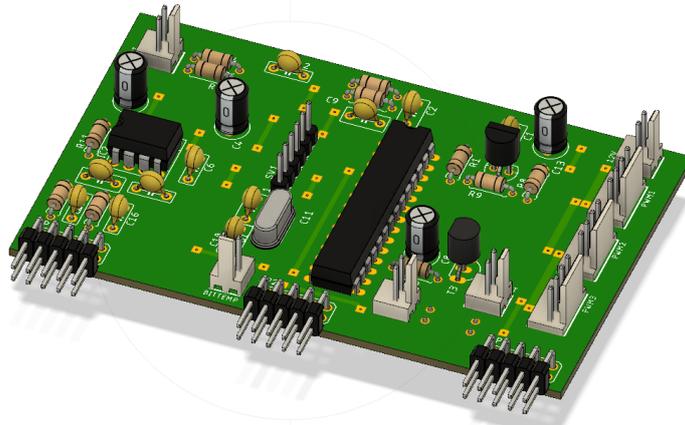
Ilustración 145 - Ilustración 136 - Señales PWM al 100% para una temperatura entre 55 y 65°C o más.

#### 4.7.12. Consumos y eficiencia

Dado que los ventiladores tienen que disponer de cierta potencia para garantizar un flujo de aire adecuado, los consumos de corriente pueden impactar en la eficiencia general del sistema completo. Es por ello, que se han elegido unos ventiladores con una demanda máxima de corriente de 0.25A. Dado que hay cuatro ventiladores, se demanda un total de 1A de la fuente auxiliar del sistema. Se pueden observar los consumos según el ciclo de trabajo de los ventiladores:

VDC (V)	IDC (A)	Ciclo de trabajo (%)	Potencia (W)
12,17	0,289	25%	3,518
12,17	0,548	50%	6,670
12,18	0,818	75%	9,968
12,16	1,09	100%	13,288

## 4.8. Placa controladora convertidor DC – DC



*Ilustración 146 - Modelo 3D controlador del convertidor DC - DC*

Para controlar el convertidor DC – DC, se necesita un dispositivo, que sea capaz de medir los valores analógicos más relevantes, y que, además, sea capaz de ejecutar un lazo de control que actúe sobre las señales PWM.

Dicho dispositivo, será capaz de comunicarse con el controlador del panel frontal para enviar información al usuario. Hoy en día, estas tareas, pueden realizarse con un solo microcontrolador especializado, es decir, un DSC (Digital Sampling Controller).

Los DSC, son dispositivos derivados de los DSP (Digital Sampling Processor) especializados en el control de procesos industriales, especialmente motores y convertidores DC – DC. Se caracterizan por compartir instrucciones con los DSP, como MAC (Multiply Accumulate), así como integrar multiplicadores y divisores hardware.

Dado que se especializan en el control, los DSC añaden módulos PWM avanzados, en los que destacan distintos tipos de sincronismo y disparo de eventos, como la conversión analógica. También permiten trabajar en distintos modos complementarios, así como añadir desfase y tiempos muertos (Dead Time) entre señales, de forma sencilla.

### 4.8.1. dsPIC33EV256GM002



*Ilustración 147 - Microcontrolador dsPIC33EV256GM002. Fuente: Microchip*

Se ha elegido como controlador principal, un DSC del fabricante Microchip modelo dsPIC33EV256GM002, que es un microcontrolador de 16 bits y funciona a 5V. Destaca

por tener un conversor A/D de 12 bits y 500 Ksps y tres timers PWM de control avanzado, que hacen un total de seis salidas PWM.

Además, tiene una gran potencia de cálculo, 70 MIPS, que le permite ejecutar algoritmos de control en cuestión de microsegundos. Dispone de multiplicadores y divisores hardware que se ejecutan en un solo ciclo de reloj. Eso sí, todas las operaciones deben realizarse en formato entero Q15, no dispone de unidad de procesamiento de coma flotante.

Está disponible en formato PDIP de 28 pines, que es adecuado para prototipar, y provee de suficientes entradas y salidas. Tiene un consumo aproximado a 60 MIPS de 30mA, que es como se ha configurado en este proyecto.

#### 4.8.2. Regulador de tensión



*Ilustración 148 - Regulador de tensión AMS1117-5.0 Fuente: Aliexpress*

Para regular la tensión de 12V de entrada a 5V en la tarjeta, como el consumo es de unos pocos mA, se ha utilizado una modernización del LM7805, el AMS1117-5.0. El motivo principal es que está disponible en paquetes SOT223, que permiten ahorrar espacio en la placa.

#### 4.8.3. Referencia de tensión



*Ilustración 149 - Referencia de tensión MCP1501. Fuente: Microchip*

Los microcontroladores dsPIC3EV, disponen de un ADC de 12 bits, pero por desgracia, no soportan referencias de tensión por debajo de  $VDD - 0.3V$ . Esta limitación impide utilizar una referencia de 4.096V como la MCP1501-40 de Microchip que se ve en la ilustración 147, que daría una sensibilidad al conversor de 1mV/bit.

Esta sensibilidad nos permitiría tener mejor precisión para leer valores analógicos. Además, esta referencia de Microchip, presenta una buena precisión de un 0.1% y una variación con la temperatura de 50 ppm.

Device Supply						
AD01	AVDD	Module VDD Supply	Greater of: $VDD - 0.3$ or VBOR	—	Lesser of: $VDD + 0.3$ or 5.5	V
AD02	AVSS	Module VSS Supply	$VSS - 0.3$	—	$VSS + 0.3$	V

*Ilustración 150 - Se observa que la tensión AVDD. no pueden ser menor que  $VDD - 0.3V$  ni mayor que  $VDD + 0.3V$ . Fuente: Datasheet dsPIC33EV*

4.8.4. Referencia utilizada: TL431A



Ilustración 151 - Referencia de tensión TL431A. Fuente: Aliexpress

Dada la imposibilidad de usar una referencia menor a 5V, se ha optado por usar el modelo TL431A del fabricante Texas Instruments, que dispone de una precisión del 1%.

Se configura con tres terminales y un divisor de tensión. La tensión a la salida se obtiene con la siguiente fórmula:

$$V_{out} = \left(1 + \frac{R1}{R2}\right) \times V_{ref}$$

En nuestro circuito queremos obtener una referencia de 2.5V, por ello, hemos utilizado  $R1 = R2 = 10k$ :

$$V_{out} = \left(1 + \frac{10k}{10k}\right) \times 2.5V = 5V$$

Se utiliza una tercera resistencia entre la entrada y la salida para limitar la corriente entre los valores aceptables de 1 y 100mA- En este caso, se ha colocado una resistencia de 330 ohmios, que permite circular una corriente de 21mA.

$$I_{ref} = \frac{12V - 5V}{330 \text{ ohmios}} = 21mA$$

El circuito completo presenta el siguiente aspecto:

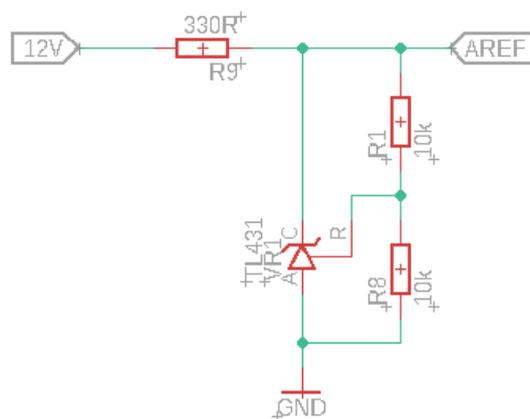


Ilustración 152 - Esquema de implementación de TL431A

#### 4.8.5. Medidas analógicas – Corriente en el primario

Para medir la corriente en el primario, se ha conectado la salida del circuito de adaptación del transformador de corriente directamente a la entrada analógica RB0.

#### 4.8.6. Medidas analógicas – Tensión en el secundario

Dado que en el secundario hay tensiones que van desde los 0V hasta 60V, deben atenuarse estos niveles de tensión a una escala de 5V con la sensibilidad suficiente para que el ADC pueda leer los cambios. Por lo que se ha utilizado un divisor de tensión formado por dos resistencias de 120k y 10k, como se observa en el apartado de Anexo – Cálculos *Lectura de la tensión en el secundario*.

#### 4.8.7. Seguidor de tensión

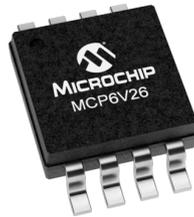


Ilustración 153 - Amplificador operacional MCP6V26. Fuente: Microchip

Para llevar estas lecturas al microcontrolador, se ha utilizado un amplificador operacional del fabricante Microchip, modelo MCP6V26, en configuración seguidor de tensión. Se ha elegido este operacional, porque tiene una tensión de offset de 2 microvoltios. Esta característica, hace que no distorsione la medida, aunque esta sea muy pequeña. Además, la tensión de salida es rail-to-rail.

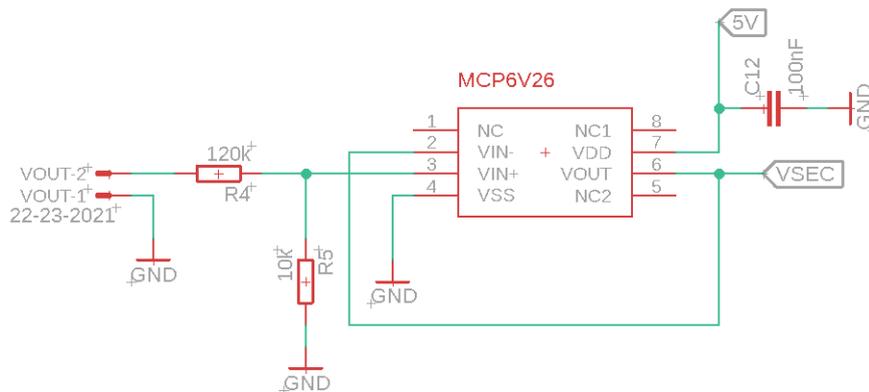


Ilustración 154 - Implementación de lectura de tensión en el secundario

#### 4.8.8. Medidas analógicas – Corriente en el secundario

Para medir la corriente del secundario, se debe leer la caída de tensión del shunt en el convertidor. Tiene un valor de 10 miliohmios, por lo que cuando circulen 16A, que es la corriente máxima de la fuente, generará una caída de 160 mV.

El shunt está posicionado en el lado de baja tensión del circuito, quedando detrás de la referencia de la fuente, por lo que la tensión que cae es negativa. Esto hace que tengamos que utilizar un amplificador operacional con alimentación simétrica bipolar, es decir,  $\pm V_{cc}$ .

Al estar el shunt en esta posición, la medida es especialmente vulnerable al modo común y un amplificador con gran rechazo a este ruido es primordial.

Por estos motivos, se ha tenido que descartar el amplificador operacional MCP6V26, que es idóneo para medidas de bajo nivel dado su bajísima tensión de offset, que no distorsionaría la lectura. Se describirán a continuación las diferentes alternativas.

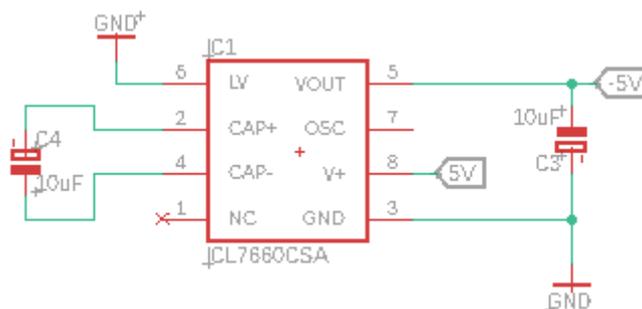
#### 4.8.9. Generador de tensión negativa: TC7660



*Ilustración 155 - Generador de tensión negativa TC7660. Fuente: Microchip*

Para generar una tensión negativa, se ha necesitado un convertidor TC7660, ya que la salida de las fuentes auxiliares es de 12V positivos. Con este dispositivo, se consigue generar una tensión negativa a partir de una tensión de entrada, en este caso +5V se convierten a -5V.

Hay circuitos especializados mejores que este, pero no se encontraban disponibles en pocas cantidades. Utilizar el TC7660 conlleva unas limitaciones que se explicarán más adelante.



*Ilustración 156 - Esquema de implementación TC7660*

4.8.10. Lectura de tensión en el shunt: Amplificador de instrumentación



Ilustración 157 - Amplificador de instrumentación INA126P. Fuente: Texas Instruments

La alternativa más empleada al medir tensiones diferenciales es, sin lugar a dudas, el amplificador de instrumentación. Por ello, se han probado varios modelos de distintos fabricantes.

El operacional finalmente empleado ha sido el INA126P de Texas Instruments, dado que, en las pruebas, al consumir solo 175 microamperios, es el que menor nivel de ruido presenta.

Esto es debido a que el TC7660, tiene una resistencia equivalente a la salida de unos 100 ohmios, por lo que se produce una caída considerable de tensión en la línea de -5V según la corriente que se le exija.

El circuito implementado con el INA126P es el representado en la ilustración 156. Una configuración de ganancia de 22.02 y dos filtros RC en las dos entradas para eliminar ruidos de alta frecuencia. Se detalla su operación en el Anexo de cálculos *Lectura de la corriente en el secundario*.

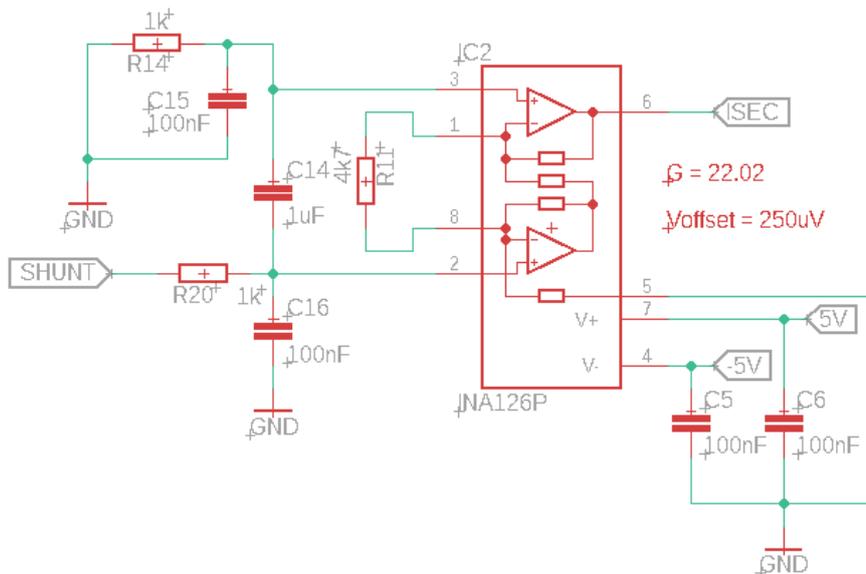


Ilustración 158 - Esquema de implementación para la lectura de corriente en el secundario

#### 4.8.11. Pruebas con otros operacionales

Se incluye un resumen de las pruebas que se han hecho con otros amplificadores.

#### AD8230 de Analog Devices

Se ha probado con este amplificador operacional de Analog Devices, pero el ruido generado por el circuito integrado TC7660 era muy grande y distorsionaba la medida cientos de mV.

Esto es debido al alto consumo de este operacional, de unos 3mA. Esto hace que la resistencia equivalente a la salida del TC7660, se haga notar:

$$\text{Variación en } -5V = Req\ TC7660 \times I_{out} = 100\ \text{ohmios} \times 3\ \text{mA}$$

$$\text{Variación en } -5V = 0.3V$$

Esto hace que la tensión caiga 0.3V, y se produzcan oscilaciones en la salida de esta magnitud. Por lo que el resultado es una medida muy afectada por el ruido.

#### AD8422BRZ de Analog Devices

Este operacional no se ha probado, pero podría ser una alternativa al INA126P, que tiene una tensión de offset muy elevado e implica un ajuste más exhaustivo.

El AD8422BRZ presenta una tensión de offset de 25 microvoltios, y al consumir solo 330 microamperios, puede que sea capaz de rechazar el ruido generado por el TC7660.

#### 4.8.12. Alternativa TC7660: ADP3605

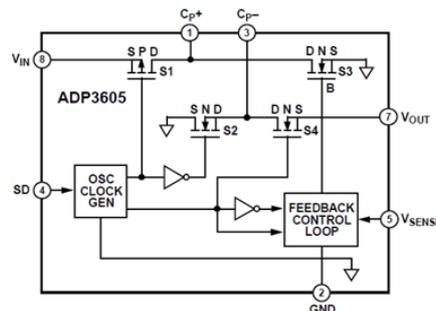


Ilustración 159 - Diagrama interno ADP3605. Fuente: Analog Devices

El ADP3605 de Analog Devices, es un dispositivo inversor de tensión, que se caracteriza por tener una salida con regulación activa, es decir, un lazo de control que mantiene el nivel de tensión en el 3% de su valor. Además, debido a este control, la resistencia de salida cae hasta los 9 ohmios y la capacidad de la salida aumenta hasta los 100mA.

Por estas razones, se considera un mejor candidato que el TC7660, con el inconveniente del precio, que es mucho mayor, y la disponibilidad.

#### 4.8.13. Puerto I2C

Para el puerto I2C, se ha implementado un conector en la tarjeta, además de dos resistencias Pull-Up de 2200 ohmios.

#### 4.8.14. Conmutación relé de salida

Para activar y desactivar la salida de la fuente a voluntad, se ha incluido una salida a un transistor 2N2222, que permita conmutar el relé de la etapa de salida.

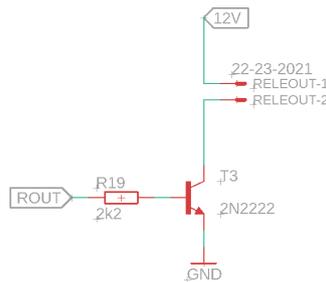


Ilustración 160 - Circuito de activación del relé de salida

#### 4.8.15. Estructura del programa

La ejecución del programa se ha dividido en varias secciones que permiten hacer que su funcionamiento sea más comprensible y no se alarguen innecesariamente los tiempos de ejecución. El programa sigue la estructura que se ve en el diagrama. (Véase el código completo en el Anexo – *Código Programa controlador DC – DC*)

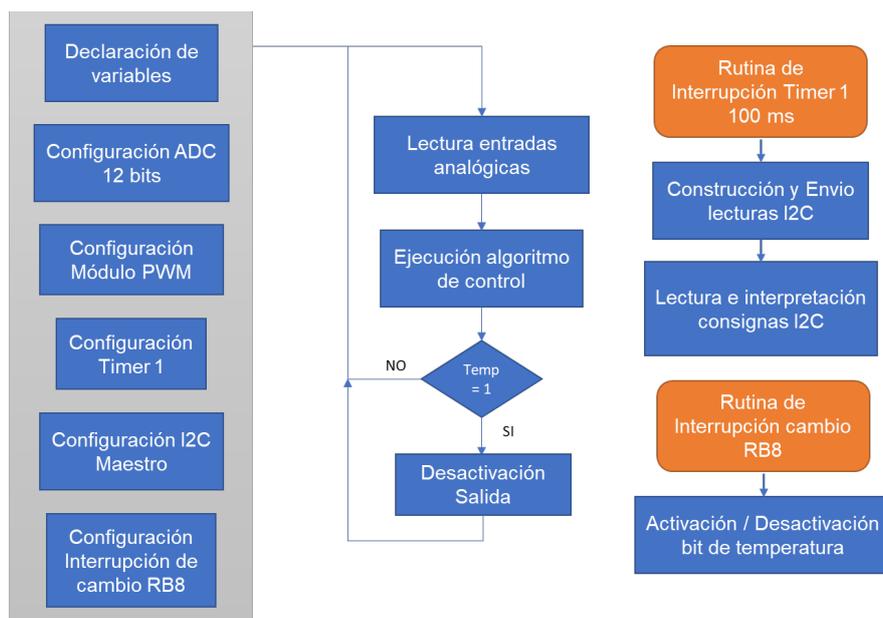


Ilustración 161 - Diagrama de flujo del programa para el controlador del convertidor DC - DC

Primero, en un archivo aparte, se han declarado todas las variables que forman parte del programa. Después se declara la configuración de los periféricos y se las distintas rutinas de interrupción. Por último, el bucle del programa principal ejecuta el algoritmo de control de forma continuada.

Una manera mejor de organizar el programa hubiera sido generar una interrupción en cada flanco ascendente de un canal PWM, por ejemplo, y ejecutar el algoritmo de control en ella para tener una periodicidad más precisa.

El ADC de este microcontrolador en el modo de 12 bits, no permite activar el muestreo automático, por lo que tenemos que iniciar todo el proceso de conversión cada vez que queramos leer las entradas.

Esto hace que el programa principal, dure unos 1000 ciclos de reloj, de los cuales 775 corresponden al proceso de conversión de las tres entradas analógicas, y 250 al algoritmo de control.

El problema es que la conversión tiene una duración temporal de:

$$\text{Tiempo de ejecución} = \frac{1}{n^{\circ} \text{ de MIPS}} \times n^{\circ} \text{ de ciclos de reloj}$$
$$\text{Tiempo de ejecución} = \frac{1}{60} \times 775 = 12.9 \text{ microsegundos}$$

Mientras que la interrupción por cada flanco ascendente de una señal PWM saltaría cada 100 KHz, es decir 10 microsegundos, por lo que no daría tiempo a terminar una interrupción, para cuando saltara la siguiente y las iría acumulando.

Por esta razón, se ha decidido dejar este proceso en el programa principal, que tiene un tiempo de ejecución aproximado de:

$$\text{Tiempo de ejecución} = \frac{1}{60} \times 1000 \text{ ciclos} = 16.66 \text{ microsegundos}$$

O lo que es lo mismo, tendríamos un tiempo de muestreo ( $T_s$ ) para el controlador de 60 KHz.

## Configuración – MCC

Para configurar los distintos periféricos, se ha utilizado la herramienta MPLAB CODE CONFIGURATOR que incluye Microchip en el entorno MPLAB X. Esto permite configurar de manera automática los periféricos que vayamos a utilizar en muy poco tiempo.

## ADC 12 bits

El ADC se ha configurado para que funcione con 12 bits y tenga el mínimo tiempo de conversión posible.

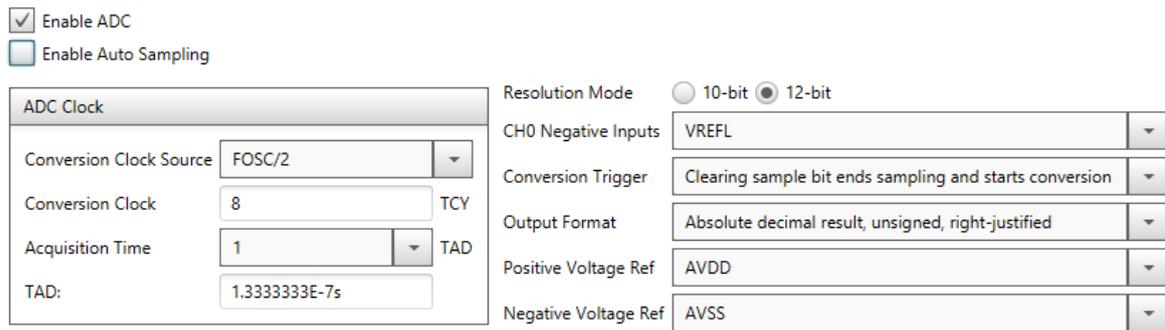


Ilustración 162 - Configuración ADC MPLAB Code Configurator

### Módulo y Salidas PWM

Las salidas PWM de los dsPIC3EV, se pueden configurar para que con un mismo timer PWM, se genere una salida como la principal, y otra como complementaria, de tal forma que la segunda sea la inversa de la primera.

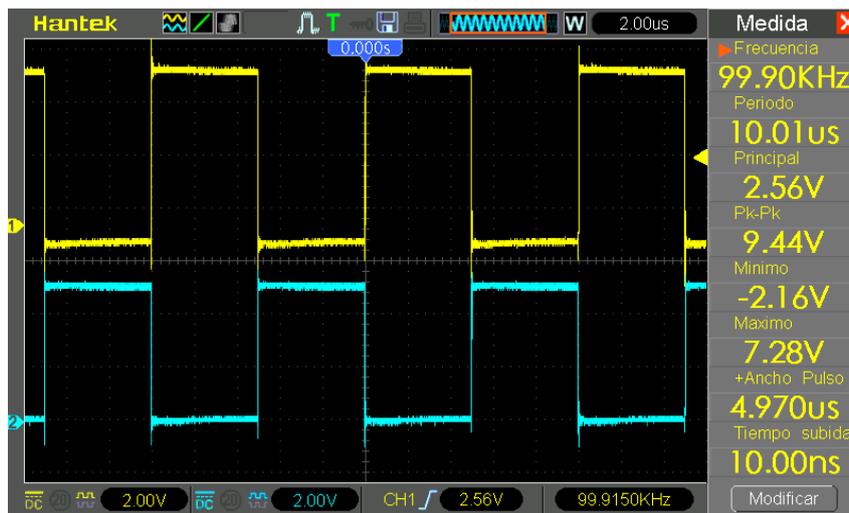


Ilustración 163 - Salidas complementarias PWM 100KHz al 50% de ciclo de trabajo

Además, para evitar la conducción cruzada de los transistores, o entrar en *hard switching* se les puede dar un tiempo muerto (Dead Time) entre ellas para que no se vaya a nivel alto la complementaria inmediatamente después de que baje la principal, sino que espere un tiempo. En este caso, se ha configurado un tiempo muerto de 120 nanosegundos.



Ilustración 164 - 120 nanosegundos aproximadamente de Dead Time

Se han utilizado en este proyecto cuatro de las seis salidas PWM disponibles. Estas salidas son las encargadas de manejar los drivers del puente completo de transistores.

Se podrían usar las dos salidas restantes para implementar una rectificación síncrona controlada, pero no entra en el alcance de este proyecto.

### Configuración I2C

El puerto I2C, se ha configurado para funcionar en modo maestro, a una frecuencia de 100KHz.

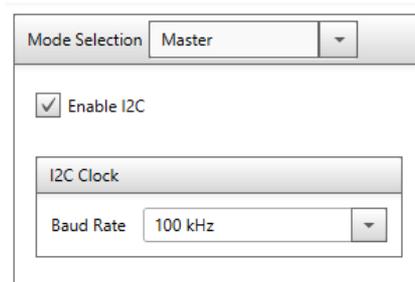
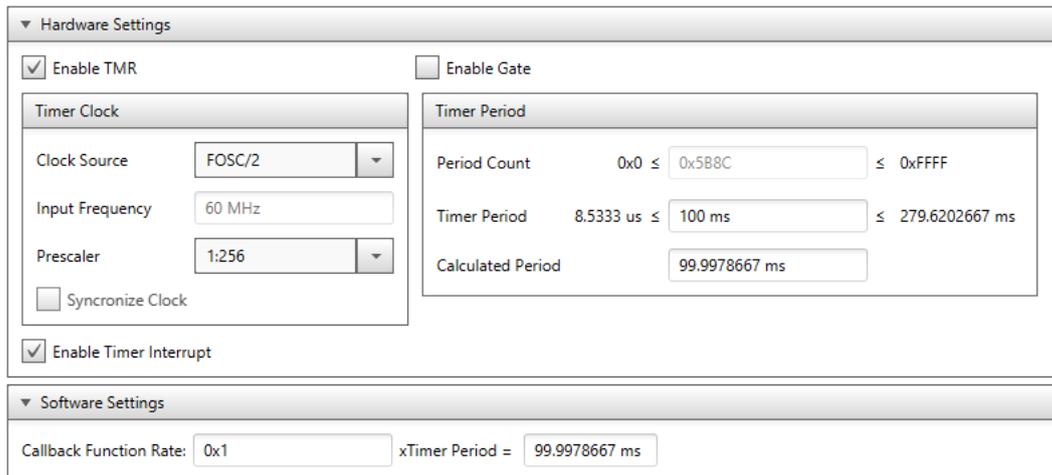


Ilustración 165 - Configuración puerto I2C Maestro MPLAB Code Configurator

### Rutinas de interrupción – Timer 1

El Timer 1 en el microcontrolador, se ha configurado para que genere una interrupción cada 100 ms. Se divide la señal de reloj por 2 antes de entrar al timer, y luego se usa una preescala de 256. Se puede ver la configuración en la imagen.



*Ilustración 166 - Configuración interrupción del timer 1 cada 100 ms MPLAB Code Configurator*

En esta rutina de interrupción, se ejecutan alternativamente dos funciones. Dichas funciones se seleccionan cambiando el valor de la variable LeerEscribir. Por ejemplo, si salta la interrupción, y LeerEscribir vale 0, se ejecuta la rutina de escritura del puerto I2C con las lecturas analógicas y el bit de temperatura.

Una vez ejecutada, la variable se cambia a 1 y cuando vuelva a saltar la interrupción, se ejecutará la lectura del puerto I2C, que leerá las consignas y el estado de la salida del controlador del panel frontal. A su vez, en esta rutina, se devuelve el valor 0 a LeerEscribir.

Esto quiere decir, que tanto las consignas, como las lecturas, se actualizan cada 200 ms.

### Rutinas de interrupción – Cambio de estado RB8

Cuando el controlador de temperatura activa la salida de sobretemperatura, se lee esa salida, en la entrada RB8 del microcontrolador principal. Para leer este valor se han activado las resistencias de Pull-Up integradas.

Los microcontroladores dsPIC33EV pueden configurarse para que cada vez que cambie el estado de una entrada, se active una interrupción. Por lo que cuando en la entrada RB8 se leen 5V, se escribe un uno en la variable Temp, y cuando se leen 0V un cero.

### Programa principal

El archivo “main.c” contiene las llamadas a la inicialización de los periféricos, así como las lecturas de los canales analógicos y la llamada al controlador principal. También se activa o desactiva la salida en función del bit de temperatura y lo recibido en el puerto I2C.

#### 4.8.16. Lazo de control por pico de corriente en el primario

Para una mejor respuesta dinámica del convertidor, se ha decidido implementar un lazo de control que varíe el ciclo de trabajo de los MOSFET, según el error de tensión obtenido (Voltage Mode Control) a través de un compensador 2P2Z.

Y que, además, compare la salida obtenida de dicho compensador a través de un DAC y un comparador, con la corriente de pico que se lee en el primario (Peak Current Control Mode). Se presenta en la siguiente imagen el sistema completo:

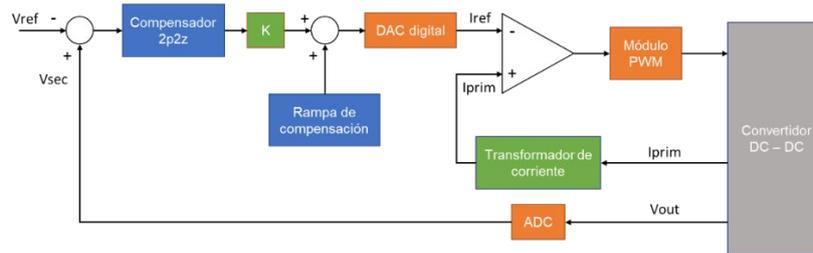


Ilustración 167 - Esquema de control a implementar. Control por pico de corriente en el primario

#### 4.8.17. Implementación controlador 2P2Z

Para implementar el controlador en el dsPIC33EV, se ha utilizado la ecuación en diferencias del controlador, con dos vectores para almacenar el error y la acción de control hasta dos muestras por detrás:

$$u(n) = B0 e(n) + B1 e(n - 1) + B2 e(n - 2) + A1 u(n - 1) + A2 u(n - 2)$$

El diseño del controlador y el cálculo de los coeficientes del mismo, son el resultado del cálculo aproximado de la planta y su operación. Al no poderse probar el convertidor, se han dejado unos coeficientes orientativos, que en fase de pruebas deberán ser modificados y ajustados para la operación correcta del dispositivo. (Véase en el Anexo – Cálculos *Cálculos controlador DC – DC*)

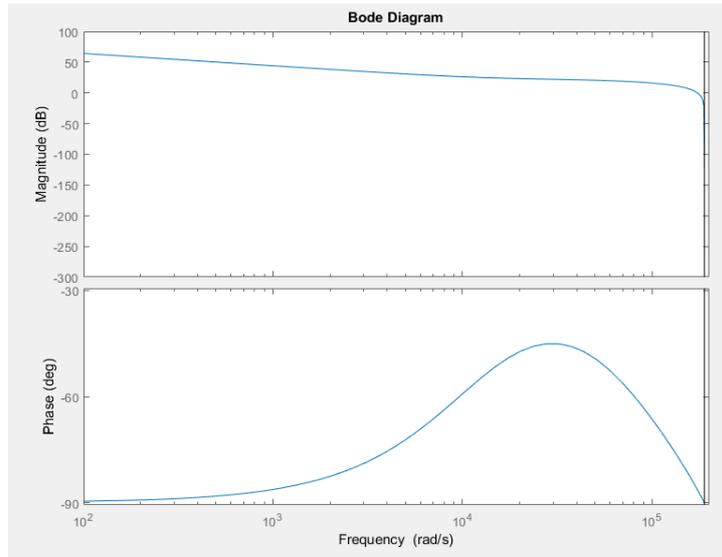


Ilustración 168 - Diagrama de Bode del controlador con  $F_s = 60000$  KHz

#### 4.8.18. Implementación con librería SMPS Control Library de Microchip

Otra manera de implementarlo, hubiera sido utilizar la librería de control especializada de Microchip, que incluye varios tipos de controladores (2P2Z, 3P3Z, PID) tanto acelerados por software, como por hardware.

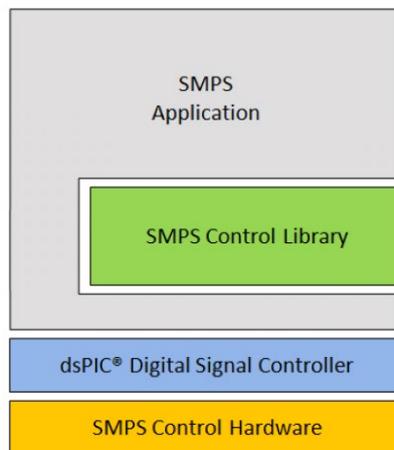


Ilustración 169 - Esquema de implementación SMPS Control Library. Fuente: Microchip

Esta librería destaca por permitir implementar un controlador, con unos tiempos de ejecución de 60 ciclos de reloj aproximadamente y según el tipo que implementemos.

Por problemas de compatibilidad con el compilador disponible, GCC, no se ha tenido tiempo de investigar su implementación en este proyecto, pero ayudaría a ahorrar gran cantidad de recursos.

#### 4.8.19. DAC y comparador digitales

Para implementar este tipo de lazos de control, los fabricantes, suelen utilizar hardware para realizar esta parte del sistema. El problema de este tipo de implementaciones, es que se está limitado por las propiedades físicas del DAC y del comparador.

Entre las ventajas de usar hardware, es que el programa es más sencillo y se tiene que hacer menos operaciones y escalas. Esto permite usar microcontroladores de menor potencia.

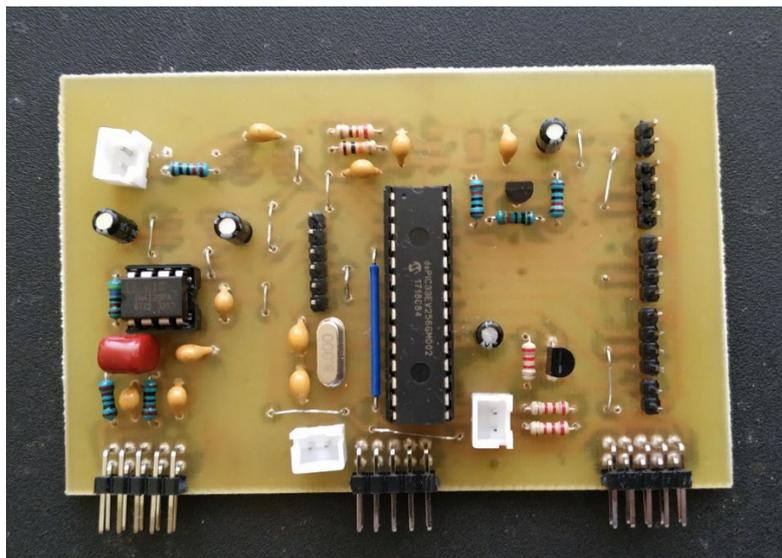
En nuestro caso, el microcontrolador dsPIC33EV, incorpora un DAC de solo 7 bits, lo que hace inviable tener una salida de tensión variable si es que se quiere tener una precisión mínima. Por lo que se ha decidido simular el DAC y el comparador en el programa.

Para esta tarea, se ha supuesto un DAC de 15 bits, para trabajar en el rango de coma fija Q15 (32768 valores) y se ha adaptado la medida de corriente del primario en el comparador en consecuencia.

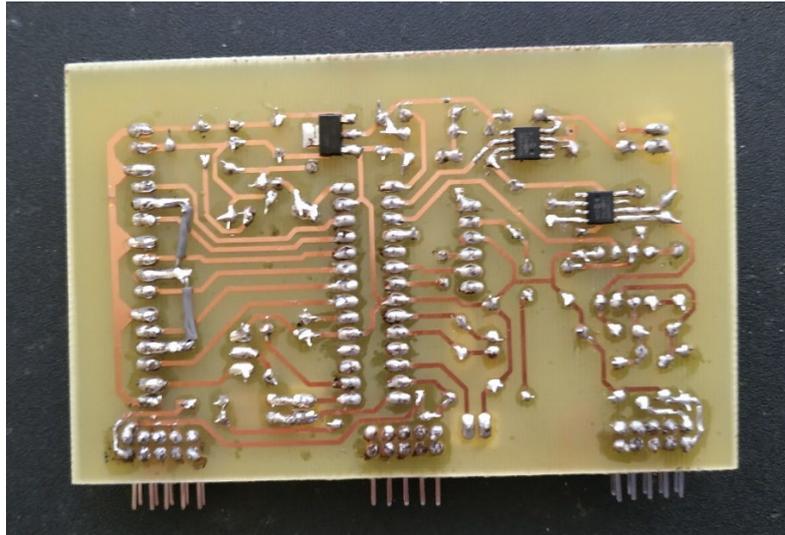
#### 4.8.20. Anti Wind-Up y saturaciones PWM

Para evitar saturaciones a la salida del controlador y ciclos de trabajo por encima de lo que el convertidor puede soportar, se ha limitado la acción de control a 32768, lo que escalado nos da un valor máximo de 576 en el módulo PWM, que equivale al 48% del ciclo de trabajo.

#### 4.8.21. Construcción

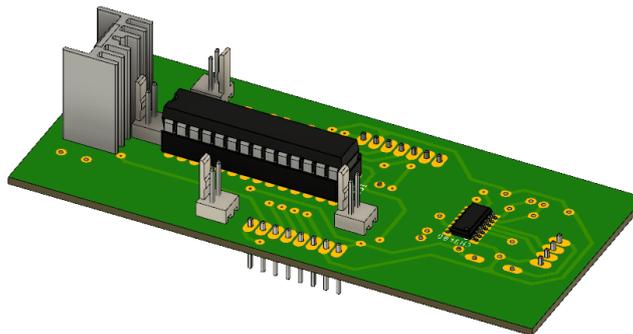


*Ilustración 170 - PCB Completa del controlador DC - DC. Vista superior.*



*Ilustración 171 - PCB Completa del controlador DC - DC. Vista inferior.*

#### 4.9. Placa de gestión del panel frontal



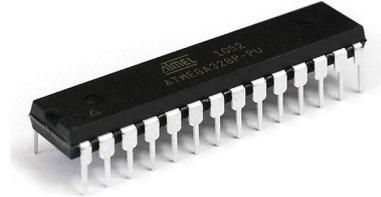
*Ilustración 172 - Modelo 3D del controlador del panel frontal*

Para recibir las señales de los botones y encoders, así como para convertirlas a consignas y comunicar las distintas partes de la fuente de alimentación y un PC, se ha desarrollado una tarjeta insertable con la botonera del panel frontal, que integra estas funcionalidades.

Consiste en un microcontrolador que lee los botones, se comunica mediante UART con la pantalla, y con I2C con el controlador principal. Además, es capaz de comunicarse por USB con un PC.

Su misión es, básicamente, la de convertir todos estos datos en cadenas que sean interpretables por dichos dispositivos, así como recibirlas y extraer datos de ellas.

#### 4.9.1. Atmega328P



*Ilustración 173 - Microcontrolador ATmega328P. Fuente: Microchip*

Para la gestión de la interfaz de usuario (pantalla y botones), así como las comunicaciones USB e I2C, se ha utilizado un microcontrolador ATmega328P del fabricante ATMEL. Es un dispositivo muy popular utilizado principalmente en las placas de desarrollo Arduino UNO y Arduino Nano.

Tiene la ventaja sobre otros procesadores de programarse en el entorno Arduino IDE, que dispone de infinidad de librerías y una gran comunidad. Estos dos motivos, permiten una programación rápida de aplicaciones sencillas.

Está disponible en formato DIP de 28 pines y tiene una potencia de cálculo de 20 MIPS. Dado que esta aplicación no requiere de tiempos de ejecución críticos, es un procesador completamente válido.

#### 4.9.2. Conversor UART – USB: CH340G



*Ilustración 174 - Conversor UART - USB CH340G. Fuente: Aliexpress*

El microcontrolador ATmega328P, no dispone de comunicaciones USB integradas, pero existen numerosos adaptadores en el mercado para traducir el protocolo UART a USB.

En este trabajo, se ha utilizado el adaptador genérico, de origen chino, CH340G, dado su bajo coste y gran disponibilidad. Es utilizado en las tarjetas de desarrollo Arduino Nano para este mismo fin, y su vez, como programador.

Tiene la ventaja de requerir muy pocos componentes para su funcionamiento:

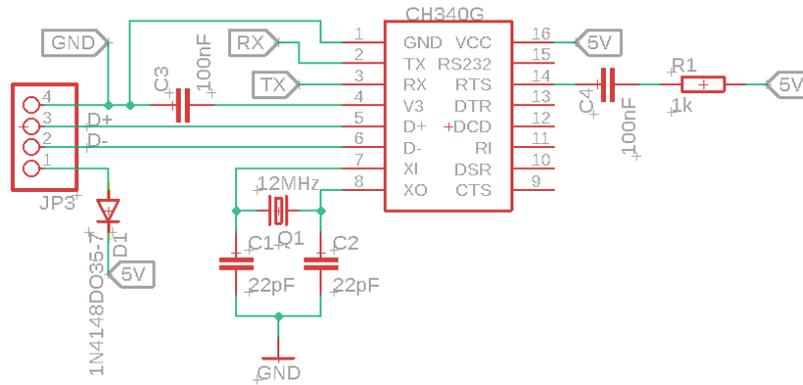


Ilustración 175 - Esquema de implementación CH340G

### 4.9.3. Alternativas CH340G



Ilustración 176 - Conversor UART - USB FTDI FT232RL. Fuente: Aliexpress

Como se ha comentado, existen innumerables alternativas, siendo los más populares, los del fabricante FTDI, pero presentan la desventaja de ser más caros y usar paquetes SMD difíciles de prototipar.



Ilustración 177 - Conversor UART - USB MCP2200. Fuente: Microchip

Otra alternativa, son la gama de adaptadores MCP2200 de Microchip. Presentan el mismo inconveniente que los de FTDI, son más caros. Tienen la ventaja de usar paquetes SOIC fácil de prototipar, pero su uso es más complejo y su disponibilidad limitada sin un pedido mínimo.

#### 4.9.4. Regulador de tensión



Ilustración 178 - Regulador LM7805. Fuente: Aliexpress

La alimentación que llega a la placa, tiene una tensión de 12V, mientras que la electrónica de ésta, funciona con 5V, por lo que es necesario colocar un regulador de tensión de 5V LM7805.

Al utilizar un regulador lineal de este tipo, debemos tener en cuenta que son muy poco eficientes, disipando gran cantidad de calor en la conversión según la corriente que hagamos circular. Podemos estimar la potencia que disipa con las siguientes fórmulas:

$$\text{Caída de tensión LM7805} = V_{in} - V_{out} = 12V - 5V = 7V$$

La pantalla TFT Nextion de 3.2 pulgadas, tiene un consumo de 150mA con el brillo al máximo, y el microcontrolador y el conversor, unos 30mA. Por lo que podemos estimar, que la placa consume 180mA en total:

$$P = 7V \times 0,18 A = 1,26W$$

Las fórmulas nos dicen que el LM7805, disipa 1.26W en forma de calor para dar energía a los circuitos y la pantalla. Por ello, se ha colocado un disipador que contenga su temperatura.

#### 4.9.5. Estructura del programa

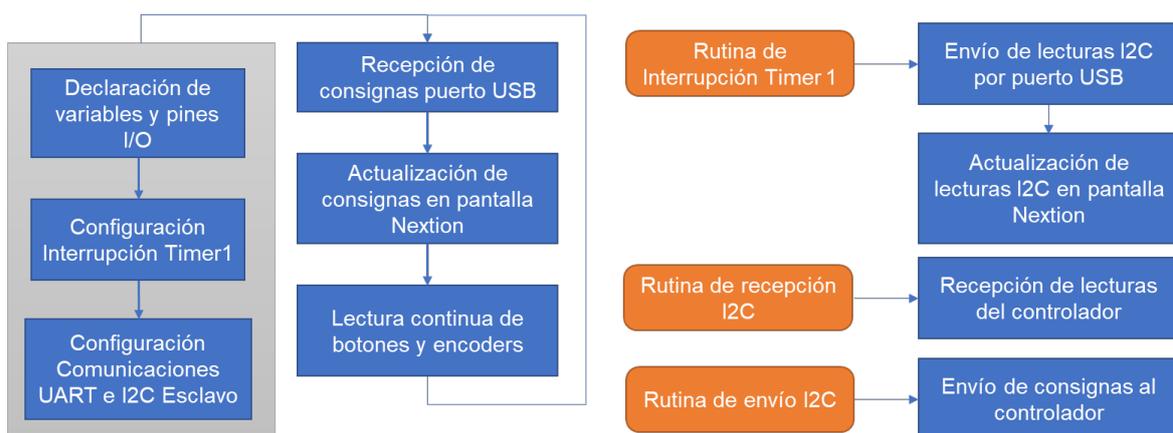


Ilustración 179 - Diagrama de flujo del programa para la gestión del panel frontal

El programa encargado de gestionar el panel frontal presenta el aspecto de la imagen superior.

#### 4.9.6. Configuración y recursos utilizados

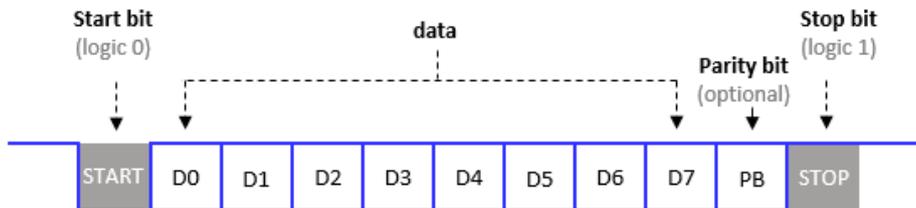


Ilustración 180 - Trama UART. Fuente: solitontech.com

Para el desarrollo del programa, ha sido necesaria la declaración de tres librerías externas. La primera, NeoSWSerial.h, es una librería que emula un puerto de comunicaciones UART por software creada por el usuario de GitHub "SlashDevin". Aunque este tipo de librería está incluida de forma oficial en la IDE de Arduino, llamada SoftwareSerial.h, está muy poco optimizada e interfiere con otros procesos del programa. Entre las ventajas de NeoSWSerial destacan:

- Menor tiempo de ejecución.
- Transmisión y recepción simultánea.
- No se desactivan las interrupciones durante todo el tiempo de recepción.
- Tiene una tasa de errores menor que SoftwareSerial.

La siguiente librería utilizada es "Nextion.h", creada por el usuario de GitHub "bborncr" como alternativa a "ITEADlib.h" del fabricante. Se ha utilizado esta librería ya que la oficial se desarrolló para usar SoftwareSerial y no permite modificarse para usar ninguna otra.

La librería de GitHub, permite utilizar cualquier medio de conexión UART, tanto software como hardware, basta con modificar unas líneas de código en el archivo de configuración de la librería.

Por último, se ha utilizado la librería "Wire.h" proporcionada por la IDE de Arduino, encargada de gestionar las rutinas de interrupción del puerto I2C.

La estructura del programa se organiza en varios segmentos, en la parte superior del programa encontramos la declaración de librerías y variables globales.

Justo debajo, en la función setup() debemos activar los pines que deseemos como salidas o entradas. Aquí se inicializan los parámetros de los puertos de comunicaciones UART, configurando los puertos hardware y software a 38400 baudios. También configuramos el puerto I2C como esclavo, y le asignamos la dirección 0x01.

Se ejecuta la rutina de inicialización de la pantalla y se precargan las cadenas de datos con los valores iniciales.

En este mismo apartado se configura el Timer 1 para interrumpir al microcontrolador cada 200 ms. Se ha elegido el Timer 1 para esta tarea porque permite contar hasta 16 bits y el Timer 0 se encarga de funciones de temporización como delay(), millis, etc en Arduino IDE, por lo que modificarlo, podría resultar conflictivo.

Primero se configura la frecuencia de reloj a la que contará el Timer 1, en este caso, seleccionamos un valor de preescalado de 1024. Esto dividirá la frecuencia del reloj principal de 16 MHz por 1024.

Table 16-5. Clock Select Bit Description

CS12	CS11	CS10	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	clk <sub>IO</sub> /1 (No prescaling)
0	1	0	clk <sub>IO</sub> /8 (From prescaler)
0	1	1	clk <sub>IO</sub> /64 (From prescaler)
1	0	0	clk <sub>IO</sub> /256 (From prescaler)
1	0	1	clk <sub>IO</sub> /1024 (From prescaler)
1	1	0	External clock source on T1 pin. Clock on falling edge.
1	1	1	External clock source on T1 pin. Clock on rising edge.

*Ilustración 181 - Tabla de preescala de reloj para el timer 1. Fuente: Datasheet ATmega328P*

Esto nos permite calcular el tiempo máximo que durará una vuelta completa del Timer 1 por sus 65536 valores:

$$\text{Tiempo máximo Timer 1} = \frac{\text{Prescaler}}{\text{Frecuencia de Reloj}} \times 2^{n^{\circ} \text{ de bits}} = \frac{1024}{16000000 \text{ Hz}} \times 65536$$

$$\text{Tiempo máximo Timer 1} = 4.19 \text{ segundos}$$

Una vez conocido el tiempo de temporización máximo del Timer 1, podemos utilizar la siguiente fórmula para averiguar qué valor tenemos que escribir en el registro de comparación para que el Timer 1 se reinicie cada 200 ms y salte la interrupción:

$$\text{Registro de comparación} = \frac{16000000 \text{ Hz}}{\text{prescaler} \times \text{Frecuencia deseada}} - 1$$

$$\text{Registro de comparación} = \frac{16000000 \text{ Hz}}{1024 \times 5 \text{ Hz (200 ms)}} - 1 = 3124$$

Para terminar, debemos activar el bit OCIE1A para habilitar las interrupciones de comparación.

Una vez configurado el Timer 1 para interrumpir el microcontrolador cada 200 ms, se escribe el código que queremos ejecutar en la rutina ISR(TIMER1\_COMPA\_vect)(). En nuestro caso, si se ha establecido comunicación con el PC, se realiza un montaje de la cadena a enviar a LabView de las lecturas del controlador principal. La cadena tiene la siguiente estructura:

$$\text{DatosParaLV} = \text{"Voltaje, Corriente, Potencia, OUT, CCCV, Temp"}$$

Se manda la tensión, corriente y potencia leídos, así como el estado de la salida (OUT), el régimen en el que trabaja la fuente (CCCV), y el bit de sobrettemperatura (Temp). En la siguiente longitud y formato:

$$\text{DatosParaLV} = "00.00,00.00,000.00,0,0,0"$$

Se mandan de esta manera, para que LabVIEW pueda distinguir unos valores de otros y se pueda hacer un seguimiento cómodo de la cadena que se envía.

Posteriormente, usando la librería Nextion.h, se envían los valores de tensión, corriente y potencia a la pantalla.

#### 4.9.7. Programa principal

El programa principal que se repite de forma periódica, se encuentra en la función loop(). Primero, se comprueba que no ha llegado nada al puerto serie con la función Serial.available(). Si ha llegado algo, se comprueba que sean las cadenas "LabViewON" y "LabViewOFF", que activan y desactivan las comunicaciones entre el microcontrolador y el PC. (Véase el código completo en el Anexo – Código *Programa Arduino Pantalla\_Botones\_USB\_I2C*)

Si se activan las comunicaciones, se activa la variable "ActivarLV" la interrupción del Timer 1 manda los valores de lectura por el puerto USB. En esta rutina, si la conexión está activa y se han recibido consignas nuevas por parte de LabView, se monta una cadena, llamada ConsignaI2C que presenta el siguiente formato y longitud:

$$\text{ConsignaI2C} = "Voltaje, Corriente, OUT"$$
$$\text{ConsignaI2C} = "00.00,00.00,0"$$

En caso de que se produzca conexión con el PC, se enviará por pantalla en los elementos "t7" y "t9", el mensaje "RmOTE" y se producirán dos pitidos, que indican al usuario que se ha conectado correctamente.

Si se produce sobrettemperatura en la fuente de alimentación, el controlador de ventiladores avisará al controlador principal y este, además de apagar la salida, alertará al procesador del panel frontal. Una vez recibida la información, se indicará por pantalla en los elementos "ON\_OFF", "t7" y "t9" que la salida está desactivada y el mensaje "TEmp".

Las rutinas siguientes, comprueban los cambios que se producen en los botones y actualizan las consignas de tensión y temperatura en cada caso en los elementos "t7" y "t9".

Si se produce una pulsación en el botón BOTON\_OUT, se actualiza la pantalla con el estado correspondiente indicando "OUT ON" y "OUT OFF" en el elemento "ON\_OFF" de la pantalla. Además, se actualiza el valor de la cadena I2C con las nuevas consignas.

A continuación, se encuentran las lecturas de los botones. Los botones se comprueban, siempre que el modo remoto no esté activado.

El botón FUNC, se utiliza para poder realizar un ajuste más preciso de las consignas. Por defecto, un giro de los encoders, produce un incremento o decremento de un valor

entero de la consigna. Es decir, si se encuentra en 0.00, un giro la incrementaría a 1.00 y así sucesivamente.

Si se pulsa el botón FUNC una vez, el incremento cambia a 0.1, y si se pulsa dos veces, a 0.01. Una tercera pulsación lo devuelve al estado inicial. De esta manera podemos ajustar en decenas de milivoltios o miliamperios.

El botón BOTON\_OUT, conmuta el estado la salida.

Los botones de los encoders, confirman los cambios en la consigna actualizando la cadena del puerto I2C.

#### 4.9.8. Puerto I2C

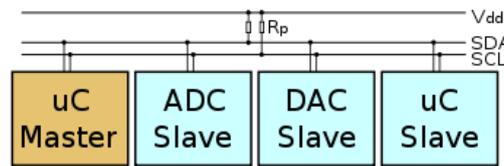


Ilustración 182 - Esquema de jerarquía protocolo I2C. Fuente: [electronicshub.org](http://electronicshub.org)

Al final del programa, se encuentran las rutinas de control del puerto I2C. La librería Wire.h, proporciona dos funciones de eventos en el puerto I2C, receiveEvent() y requestEvent(). En la primera, se hace una lectura de los bits que se reciben por el puerto y se guardan en la cadena "CadenaLecturaI2C". En la segunda, escribimos las consignas que hayamos seleccionado. En la función setup(), se declara la dirección de esclavo de este dispositivo como 0x01h.

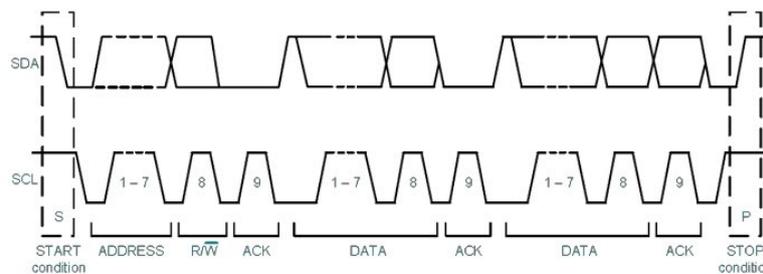


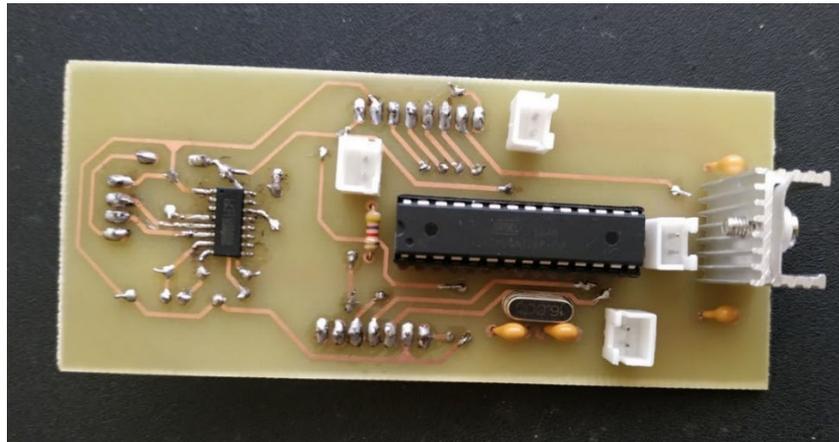
Ilustración 183 - Protocolo I2C. Fuente: [Sparkfun](http://Sparkfun)

Se ha elegido temporizar las consignas y lecturas cada 100 milisegundos, debido a la longitud que presentan las cadenas. A 100 KHz, mandar 15 bytes de datos más otro segmento para la dirección, supone un tiempo aproximado de:

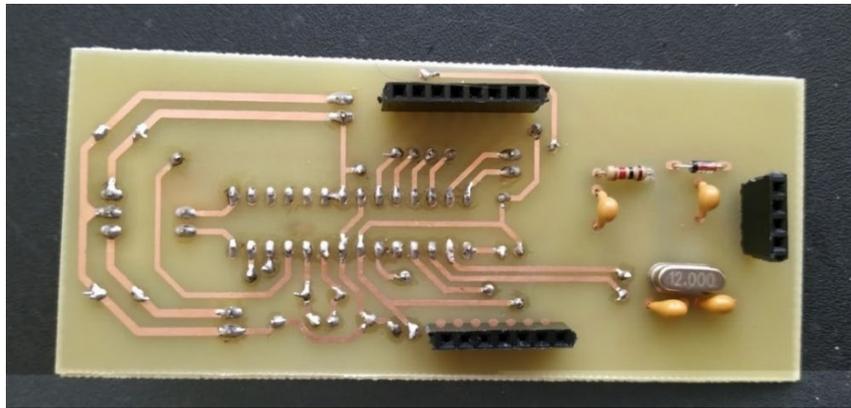
$$T_{\text{Envío I2C}} = \frac{\text{Longitud (Bytes)}}{\text{Tasa de señalización}} = \frac{16 \text{ bytes} \times 8 \text{ bits}}{100 \text{ kHz}} = 1.5 \text{ ms}$$

Por lo que cada cadena que se envía interrumpe unos 15 milisegundos cada segundo al microcontrolador encargado del convertidor principal, en un total de 10 interrupciones. Para no saturarlo, se ha decidido limitar la actualización de las lecturas.

#### 4.9.9. Construcción

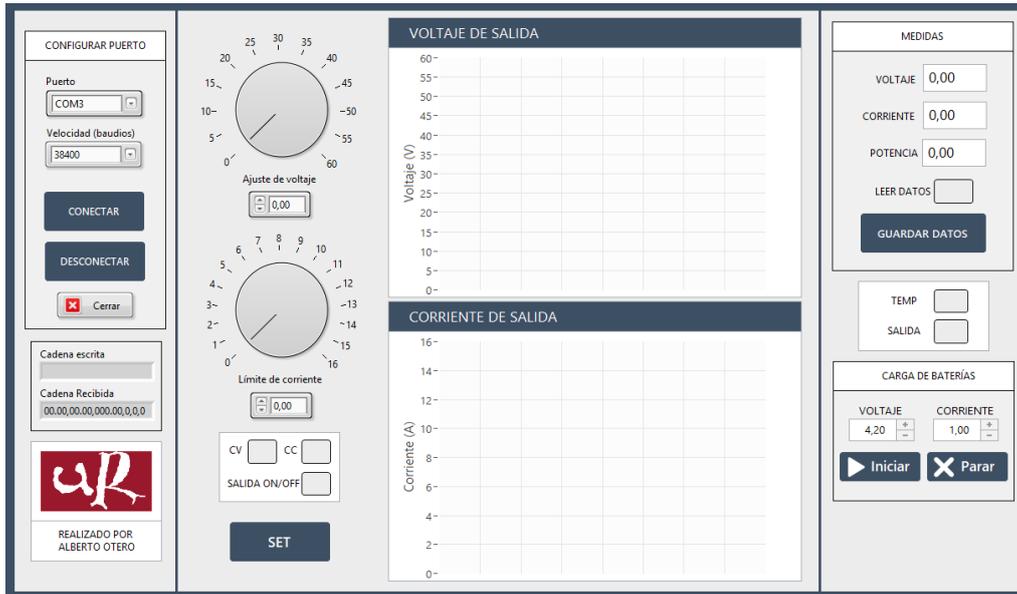


*Ilustración 184 - PCB Completa control del panel frontal. Vista superior*



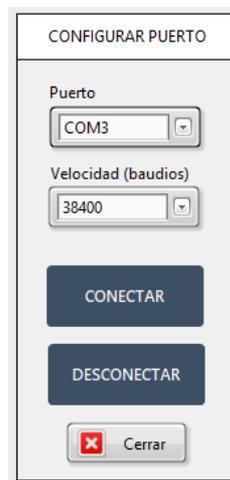
*Ilustración 185 - PCB Completa control del panel frontal. Vista inferior*

## 4.10. Control bajo PC



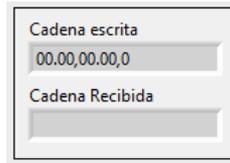
*Ilustración 186 - Interfaz completa del programa en LabView*

La aplicación para el manejo de la fuente de alimentación bajo PC, se ha desarrollado en LabVIEW, y consiste en una interfaz que permite al usuario conectarse y desconectarse de la fuente, mandar consignas, visualizar las lecturas de forma gráfica, guardar los datos en un fichero Excel, así como iniciar y detener una carga de baterías. (Véase el código completo en el Anexo – Código *Programa Control remoto fuente de alimentación LabVIEW*)



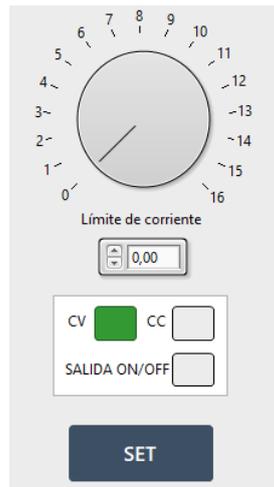
*Ilustración 187 - Gestión del puerto USB*

En la parte superior izquierda, nos encontramos con el apartado “Configurar puerto”, que permite elegir el puerto en el que se ha conectado la fuente, seleccionar la velocidad de transmisión, (por defecto 38400 baudios), así como conectarse, desconectarse y cerrar la aplicación.



*Ilustración 188 - Cadenas escritas y recibidas para depuración*

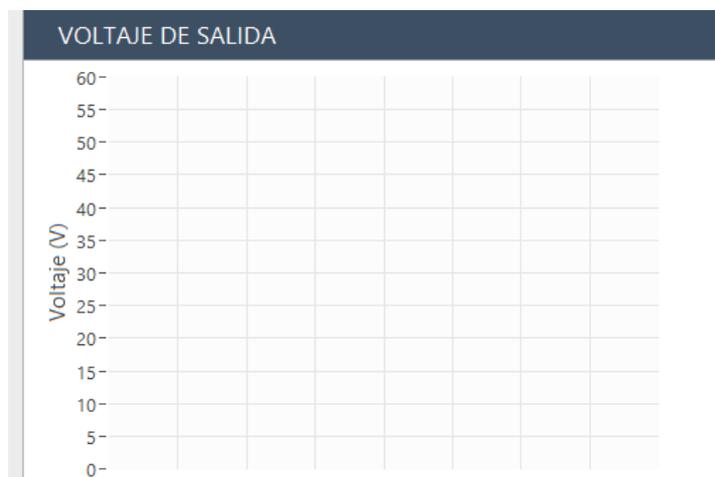
Justo debajo, nos encontramos unos cuadros de texto, que nos informan de las tramas que se están enviando en cada momento. Pueden utilizarse como herramienta de diagnóstico.



*Ilustración 189 - Controles de consigna*

A la derecha, se encuentran dos potenciómetros para ajustar las consignas de tensión y corriente. También permiten introducir los valores por teclado. Debajo se encuentran dos indicadores llamados CC y CV que informan del régimen en el que está trabajando la fuente.

En el mismo apartado, podemos observar un botón llamado SALIDA ON/OFF, que nos sirve para activar o desactivar el relé que conmuta la salida de la fuente. Por último, el botón SET sirve para aplicar la configuración que hayamos hecho en este apartado.



*Ilustración 190 - Gráficas de tensión y corriente*

Al lado de este último apartado, se encuentran las gráficas para visualizar las lecturas de tensión y corriente que hace la fuente en tiempo real.



*Ilustración 191 - Indicadores de tensión, corriente y potencia*

En la parte superior derecha, se encuentra una pestaña llamada “MEDIDAS”, en la que se muestran los valores en tiempo real de tensión, corriente y potencia, así como un botón llamado LEER DATOS que desactiva las gráficas y otro botón llamado GUARDAR DATOS que comienza la generación de un archivo Excel.

Se observan debajo, los indicadores TEMP y SALIDA, que muestran el estado de la alarma de temperatura de la fuente y si la salida está activada.

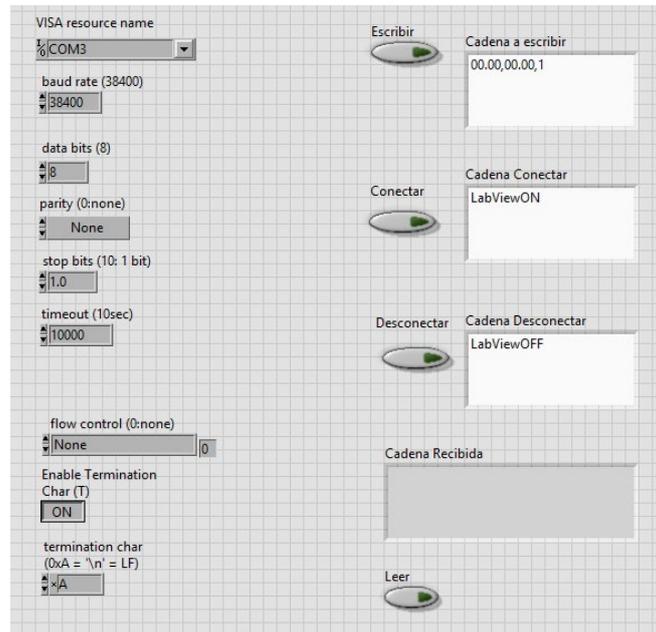


*Ilustración 192 - Módulo de carga de baterías*

Para terminar, en la parte inferior derecha, se encuentran los controles para la carga de baterías. Podemos especificar un valor de tensión, y otro de corriente para que sean las consignas máximas de la fuente, además de iniciar y parar la carga cuando deseemos.

Una vez se haya iniciado la carga, se deshabilitarán los controles de consignas, y se deberá esperar a que termine la carga o pulsar el botón “Parar”. Cuando termina la carga, aparece un mensaje en una ventana emergente “Carga completada”.

#### 4.10.1. Control bajo PC - Bloque Gestión Puerto USB



*Ilustración 193 - Panel frontal del bloque de gestión USB*

Para el control del puerto USB, se ha utilizado la librería NI VISA – Serial, en la que se configuran el número de puerto, la velocidad, paridad, etc. Se han utilizado controles en forma de botón para manejar la lectura y escritura del puerto.

Se ha establecido para conectarse con la fuente, el envío de la cadena “LabViewON” y para desconectarse, la cadena “LabViewOFF”.

## 5. Conclusiones

En este proyecto, se ha desarrollado completamente una solución técnica viable para la construcción de una fuente de alimentación programable. Se han fabricado prototipos que representan las diferentes partes que podrían formar parte de un sistema funcional.

Se han realizado diversas pruebas que así lo demuestran (**Referencia Apartado anexos: pruebas**) pero finalmente, por un error humano, se han dañado algunos componentes del convertidor PFC, y no se han podido realizar pruebas de eficiencia ni funcionamiento, al no disponer de los materiales necesarios para su reparación con el margen de tiempo suficiente.

Se ha realizado el diseño y la fabricación del convertidor DC – DC ZVS de puente completo, así como la implementación hardware y la programación de su controlador, pero se dejan los coeficientes del controlador 2P2Z tal y como los arrojan los cálculos a falta de probar en un modelo real y realizar el ajuste final de la instrumentación.

Se ha desarrollado por completo, el sistema de control de temperatura y refrigeración, así como los drivers de potencia para los MOSFET del convertidor DC – DC.

También se ha finalizado, el hardware de gestión del panel frontal y las comunicaciones I2C con la placa principal y UART con la pantalla y el ordenador. De tal forma, que funcionan la interfaz de usuario y la selección de consignas, así como el software de control bajo PC, que es capaz de controlar el sistema y de generar ficheros de datos con las lecturas que se realizan.

Además del diseño de la electrónica y el software del sistema, se ha realizado el diseño mecánico de la fuente, como se ha podido ver a lo largo de la memoria, en el que se incluyen los paneles de interfaz de usuario de plástico, así como la carcasa exterior en acero y modelos 3D de todas las placas de circuitos realizadas para comprobar que, efectivamente, cumplan con las dimensiones requeridas.

Firmado, a 31 de agosto de 2019, Logroño



Alberto Otero Olavarrieta



# ANEXO I: CÁLCULOS



## Índice

1.	Dimensionamiento filtro EMI .....	133
1.1.	Atenuación del ruido en modo diferencial .....	133
1.2.	Atenuación del ruido en modo común.....	134
2.	Dimensionamiento del rectificador.....	134
3.	Dimensionamiento Convertidor Boost PFC.....	134
3.1.	Dimensionamiento de la bobina .....	134
3.2.	Dimensionamiento del transistor MOSFET .....	135
3.3.	Dimensionamiento del diodo Boost .....	136
3.4.	Dimensionamiento condensador de filtro del convertidor .....	136
4.	Dimensionamiento convertidor DC – DC.....	137
4.1.	Dimensionamiento MOSFET puente completo .....	137
4.2.	Diseño Transformador de la fuente conmutada .....	137
4.3.	Dimensionamiento Bobina de resonancia LR .....	140
4.4.	Dimensionamiento MOSFET rectificación síncrona .....	141
4.5.	Diseño etapa de filtrado en el secundario .....	141
5.	Dimensionamiento del sistema de refrigeración de la fuente.....	142
5.1.	Incremento de temperatura en el rectificador y los componentes del convertidor Boost PFC.....	142
5.2.	Incremento de temperatura en los MOSFET del convertidor DC – DC .....	143
5.3.	Disipador MOSFET rectificación síncrona.....	143
5.4.	Dimensionamiento MOSFET de control de ventiladores .....	144
6.	Cálculos de escala de medidas analógicas .....	144
6.1.	Lectura de la tensión en el secundario .....	144
6.2.	Lectura de la corriente en el secundario.....	145
7.	Cálculos controlador DC – DC .....	145
7.1.	Escalado lecturas analógicas .....	145
7.2.	Cálculo del controlador 2P2Z .....	146
7.3.	Cálculo de la rampa de compensación digital .....	148



## 1. Dimensionamiento filtro EMI

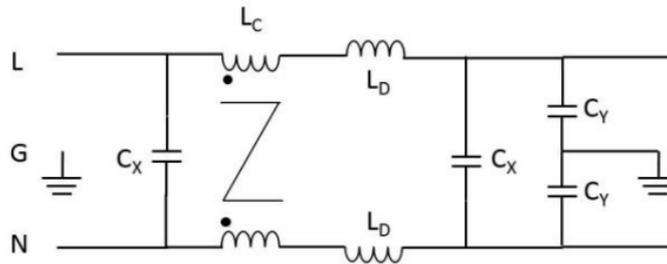


Ilustración 194 - Esquema básico filtro EMI

Para calcular un filtro EMI, se divide la metodología de trabajo en las categorías figuradas en el siguiente diagrama:

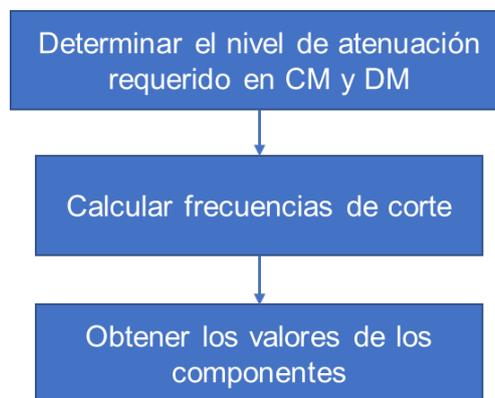


Ilustración 195 - Procedimiento de cálculo de filtros EMI

### 5.1. Atenuación del ruido en modo diferencial

Primero, debemos determinar la frecuencia de corte de nuestro filtro, en modo común y un modo diferencial.

Para determinar la frecuencia de corte en modo común y sus componentes utilizamos la siguiente expresión:

$$f_{R,CM} = \frac{1}{2\pi\sqrt{Lc \times 2 \times Cy}}$$

Suponemos:  $f_{R,CM} = 25\text{KHz}$

$Cy = 4700 \text{ pF}$  debe ser menor de  $5400\text{pF}$  para evitar verter corrientes de fuga a tierra

Obtenemos:  $Lc = 4,31 \text{ mH}$

Dado que no existe en el mercado una bobina de modo común del valor exacto, seleccionamos una bobina del valor inmediatamente superior, 5 mH.

## 5.2. Atenuación del ruido en modo común

Para determinar la frecuencia de corte en modo diferencial y sus componentes:

$$f_{R,CM} = \frac{1}{2\pi\sqrt{((2 \times L_D) + L_{Leak}) \times Cx}}$$

Suponemos:  $f_{R,DM} = 10\text{KHz}$

$$L_{DM} = 100 \mu\text{H}$$

$$L_{Leak} = 0,01 \mu\text{H}$$

Obtenemos:  $Cx = 1,27 \mu\text{F}$

## 2. Dimensionamiento del rectificador

A continuación, se detallan los cálculos necesarios para dimensionar correctamente el rectificador en nuestra aplicación:

$$I_{media} = \frac{2}{\pi} \times \frac{\sqrt{2} \times P_{out}}{VAC_{LL}} = \frac{2}{\pi} \times \frac{\sqrt{2} \times 1000 \text{ W}}{207 \text{ VAC}} = 4,35 \text{ A}$$

$$P_{puente} = 2 \times I_{media} \times V_{f \text{ puente}} = 2 \times 4,35 \text{ A} \times 1 \text{ V} = 8,69 \text{ W}$$

## 3. Dimensionamiento Convertidor Boost PFC

### 3.1. Dimensionamiento de la bobina

Una vez hemos definido la potencia y la tensión mínima de entrada, podemos calcular la corriente máxima que va a soportar la bobina:

$$I_{bobina, \text{máx}} = \frac{\sqrt{2} \times P_{out}}{n \times VAC_{LL}} = \frac{\sqrt{2} \times 1000 \text{ W}}{0,93 \times 207 \text{ VAC}} = 7,35 \text{ A}$$

En el caso de que nos conectáramos a la red eléctrica de un país con 120 VAC. La tensión mínima es de 108 VAC

$$I_{bobina, \text{máx}} = \frac{\sqrt{2} \times P_{out}}{n \times VAC_{LL}} = \frac{\sqrt{2} \times 1000 \text{ W}}{0,93 \times 108 \text{ VAC}} = 14,08 \text{ A}$$

Para satisfacer las condiciones de diseño de la bobina en la red eléctrica de 120VAC, se necesitaría una bobina que no se ha adquirido, ya que tenía un precio demasiado alto. Se recomienda instalar una bobina con capacidad para resistir 20A como mínimo.

Para la implementación del prototipo, todos los cálculos se han realizado para satisfacer las condiciones de diseño de la red eléctrica española de 230 VAC.

Se calcula, a continuación, el valor necesario de inductancia suponiendo un rizado de la corriente de la bobina de un 25%:

$$L = \frac{1}{\% \text{ rizado}} \times \frac{VAC_{min}^2}{Pout} \times \left(1 - \frac{\sqrt{2} \times VAC_{min}}{Vout}\right) \times T$$

Sustituimos los valores:

$$L = \frac{1}{0.25} \times \frac{(207 VAC)^2}{1000W} \times \left(1 - \frac{\sqrt{2} \times 207}{400 V}\right) \times \frac{1}{100000 Hz} = 460 \mu H$$

Se debe considerar, para utilizar este valor, que, según el material del núcleo utilizado en la bobina, puede perder su inductancia entre un 75 y un 80%. Por lo que para mantener este nivel de inductancia a máxima potencia debemos dimensionar la bobina teniendo en cuenta esa consideración. Resultado en una bobina de:

$$L = \frac{460 \mu H}{25\%} = 2 mH$$

### 3.2. Dimensionamiento del transistor MOSFET

Calculamos la corriente RMS a través de la línea de 50Hz que debe soportar:

$$I_{S,rms} = \frac{Pout}{VAC_{min}} \times \sqrt{1 - \frac{8 \times \sqrt{2} \times VAC_{min}}{3 \times \pi \times Vout}} = \frac{1000 W}{207 VAC} \times \sqrt{1 - \frac{8 \times \sqrt{2} \times 207 VAC}{3 \times \pi \times 400 V}} = 2.97 A$$

Ahora podemos calcular las pérdidas que se producen en la conducción, asumiendo que  $R_{ds}(ON, 100^\circ C)$  es  $R_{ds}(ON, 25^\circ C)$  x 1.8:

$$P_{S,cond} = I_{S,rms}^2 \times R_{on(100^\circ C)} = (2,97 A)^2 \times 0,27 \Omega \times 1.8 = 4,29 W$$

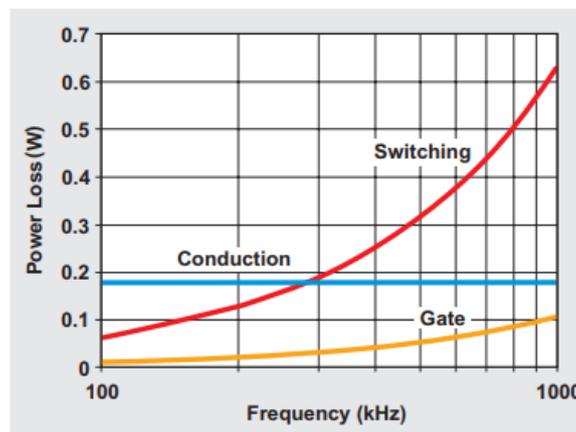


Ilustración 196 - Pérdidas MOSFET Conducción VS Conmutación.

Debemos tener en cuenta las pérdidas que se producen en las conmutaciones del MOSFET, es decir, en el tiempo de encendido y apagado. Como puede verse en la figura, a mayor frecuencia, más importante se hacen las pérdidas de este tipo. Las pérdidas de conmutación pueden estimarse con la siguiente fórmula:

$$P_{sw} = \frac{2}{3} \times C_{oss} \times \sqrt{25} \times V_{on}^{1.5} \times f_{sw} = \frac{2}{3} \times 870 \times 10^{-12} \times 5 \times 400^{1.5} \times 100000 = 2.32 \text{ W}$$

Por lo que las pérdidas totales en el MOSFET son de:

$$Perdidas \text{ MOSFET} = P_{S,cond} + P_{sw} = 4.29 \text{ W} + 2.32 \text{ W} = 6.61 \text{ W}$$

### 3.3. Dimensionamiento del diodo Boost

Para dimensionar el diodo que transmite la corriente a la salida del convertidor, se pueden utilizar los siguientes cálculos:

$$I_{D,avg} = \frac{P_{out}}{V_{out}} = \frac{1000 \text{ W}}{400 \text{ V}} = 2.5 \text{ A}$$

Una vez que hemos determinado la corriente que circula por el diodo, podemos calcular las pérdidas de conducción, suponiendo una tensión de conducción de 1,2V:

$$P_{D,cond} = I_{D,avg} \times V_{f,diodo} = 2.5 \text{ A} \times 1,2 \text{ V} = 3 \text{ W}$$

Estas serán las pérdidas que se producen en la conducción del diodo, pero también se producen pérdidas en las conmutaciones que le induce el MOSFET con la carga entre la puerta y la fuente (Qgs):

$$P_{D,sw} = 0.5 \times V_{out} \times Q_{gs} \times freq = 0.5 \times 400 \text{ V} \times 29 \times 10^{-9} \times 100000 \text{ Hz} = 0,58 \text{ W}$$

Por lo tanto, las pérdidas totales en el diodo Boost son:

$$P_{D,cond} + P_{D,sw} = 3 \text{ W} + 0,58 \text{ W} = 3.58 \text{ W}$$

### 3.4. Dimensionamiento condensador de filtro del convertidor

El condensador de salida es el encargado de mantener el tiempo de Hold-Up (16.6ms) y de mantener un rizado de baja frecuencia de 10V. El condensador que se seleccione deber tener una capacidad mayor que la resultante de la siguiente fórmula:

$$C_o > \frac{P_{out}}{2 \times \pi \times freq \text{ línea} \times V_{rizado} \times V_{out}} = \frac{1000 \text{ W}}{2 \times \pi \times 50 \text{ Hz} \times 10 \text{ Vpp} \times 400 \text{ V}} = 795,8 \text{ uF}$$

Debemos tener en cuenta la corriente RMS que circula a través de la línea de 50Hz con la siguiente ecuación:

$$I_{Co,rms} = \sqrt{\frac{8 \times \sqrt{2} \times P_{out}^2}{3 \times \pi \times V_{AC,min} \times V_{out}} - \frac{P_{out}^2}{V_{out}^2}} = \sqrt{\frac{8 \times \sqrt{2} \times 1000^2}{3 \times \pi \times 207 \text{ VAC} \times 400 \text{ V}} - \frac{1000 \text{ W}^2}{400 \text{ V}^2}}$$

$$I_{Co,rms} = 2,87 \text{ A}$$

Ahora se debe calcular la potencia que disipan los condensadores por la presencia de dicha corriente:

$$P_{Co} = I_{Co,rms}^2 \times ESR = 2,87^2 \times 0,29 = 2,39 \text{ W}$$

## 4. Dimensionamiento convertidor DC – DC

### 4.1. Dimensionamiento MOSFET puente completo

Los esfuerzos que debe soportar vienen dados por las siguientes fórmulas:

Primero calculamos la capacidad media de salida de los transistores, siendo  $C_{oss}$  y  $V_{ds}$  especificadas en la hoja de datos:

$$C_{oss \text{ avg}} = C_{oss} \times \sqrt{\frac{V_{ds}}{V_{in \text{ max}}}} = 870 \times 10^{-12} \times \sqrt{\frac{25V}{400V}} = 190 \text{ pF}$$

También se especifica la carga en la puerta del transistor  $Q_g$ . Y basándonos en la tensión que aplicaremos para activarlo ( $V_g = 12V$ ), podemos calcular las pérdidas de conducción y conmutación:

$$P_{\text{pérdidas MOSFET}} = I_p^2 \times R_{ds} (ON) + 2 \times Q_g \times V_g \times \frac{f_{sw}}{2}$$

$$P_{\text{pérdidas MOSFET}} = \left( \frac{1000W}{380V} \times \sqrt{2} \right)^2 \times 0,27 \text{ ohmios} + 2 \times 29 \times 10^{-9} \text{ C} \times 12V \times \frac{100 \text{ KHz}}{2}$$

$$P_{\text{pérdidas MOSFET}} = 3,77W$$

En realidad, se produce la conducción simultánea de dos de los transistores cada vez, por lo que podemos estimar unas pérdidas de 7.54W. Pero como todos contribuyen al incremento de temperatura en el disipador, se hace una estimación de:

$$P_{\text{pérdidas totales MOSFET}} = 3,77W \times 4 = 15,09W$$

### 4.2. Diseño Transformador de la fuente conmutada

Para diseñar un transformador para una fuente conmutada, debemos seguir una serie de pasos:

1. Definir los parámetros eléctricos del transformador, esto es:

$$V_o = 60V \quad I_o = 16A \quad P_o = 1000 \text{ W}$$

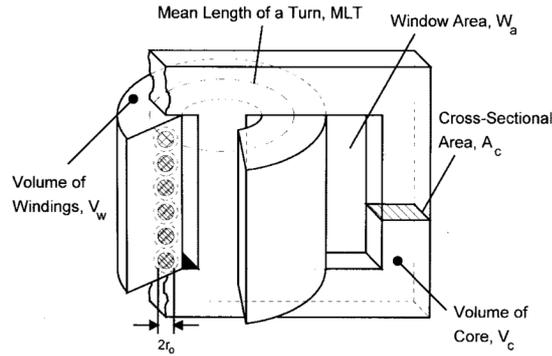


Ilustración 197 - Esquema de los parámetros de un transformador

2. Calcular el producto del área (Ap) del área de ventana (Aw) y el del núcleo (Ac). Definida por la siguiente fórmula:

$$A_p = A_c \times A_w = \frac{P_o \times (1 + \frac{1}{\eta})}{\sqrt{2} \times k_w \times J \times f_{sw} \times B_m}$$

Donde  $\eta$  es la eficiencia estimada, se supondrá un 80%.  $k_w$  es el factor de apilamiento del bobinado, con un valor de 0.4.  $J$  es la densidad de corriente deseada, generalmente entre 4 y 6 A/mm<sup>2</sup>. Y  $B_m$  es la magnetización del núcleo en teslas, en este caso 0.3T.

Sustituyendo:

$$A_p = \frac{1000W \times (1 + \frac{1}{0.8})}{\sqrt{2} \times 0.4 \times 4 \times 100000 \times 0.3} = 0.033 \text{ m}^2$$

Una vez hemos estimado un valor para  $A_p$ , podemos comparar directamente este valor, con el producto de  $A_w$  y  $A_c$  del núcleo que elijamos. El área del transformador siempre debe ser mayor que este número.

Comparándolo con el área central de un núcleo EE55/28/21

$$A_{p \text{ núcleo}} = (17 \text{ mm} \times 21 \text{ mm}) \times (19 \text{ mm} \times 10 \text{ mm} \times 2) = 135660 \text{ mm}^2 = 0.13566 \text{ m}^2$$

$$0.13566 \text{ m}^2 > 0.033 \text{ m}^2$$

Por lo que cumple con la especificación de área. Se ha elegido este núcleo para diseñar el transformador dada su disponibilidad en bajas cantidades y bajo precio. De hecho, como se puede observar en esta cifra y en las siguientes, este núcleo está muy sobredimensionado, pero para acceder a núcleos más pequeños y eficientes como el PQ4040, se debe realizar un pedido mínimo.

3. Calculamos el número de vueltas en el primario  $N_p$ :

$$V_{in} = 2 \times N_p \times A_c \times B_m \times f_{sw}$$

$$400V = 2 \times N_p \times (17 \text{ mm} \times 21 \text{ mm}) \times 0.3T \times 100000 \text{ Hz}$$

$$N_p = 18.67 \text{ vueltas en el primario}$$

Se deciden bobinar 18 vueltas en el primario.

4. Calculamos el número de vueltas en el secundario:

$$\frac{N_s}{N_p} = n = \frac{60V}{400V}$$

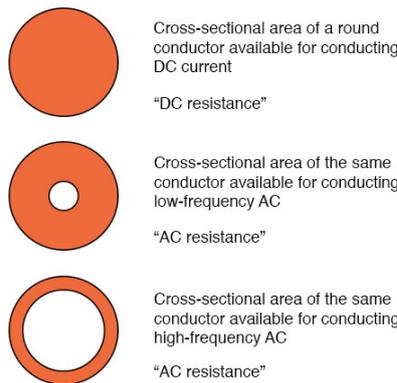
$$N_s = 2.7 \text{ vueltas en el secundario}$$

Se deciden bobinar 4 vueltas en el secundario para tener cierto margen de regulación.

5. Se calcula el área que ocupan los bobinados en la ventana del transformador.

Dado que la electricidad a partir de cierta frecuencia deja de conducirse por el centro de los cables y se va a hacia los laterales, se produce el denominado “efecto piel”. Eso hace que por el centro de la sección no circula prácticamente corriente, sino que las cargas fluyen por la zona más superficial de la sección.

En este proyecto, la frecuencia de conmutación elegida es de 100 KHz, por lo que el diámetro útil máximo de los cables es de 0.4 mm según el estándar AWG, esto es producido por el ya mencionado efecto piel.



*Ilustración 198 - Explicación del efecto piel en un conductor*

Dado que conocemos la cantidad de corriente que debe circular por el primario y el secundario, se pueden colocar varios cables de un valor cercano a este diámetro en paralelo.

En la construcción del transformador, se ha usado cable de cobre de 0.45 mm capaz de soportar 2.7A aproximadamente. Dado que no conviene que el transformador se caliente demasiado, podemos colocar más cables de los necesarios. en paralelo. De esta manera, se han bobinado:

	Nº de vueltas	Nº de cables	Capacidad (A)
Primario	18	4	10.8
Secundario	4 + 4	14	37.8

Una vez conocemos el número de cables que hay en el transformador, podemos calcular el área que ocupan:

$$awp = \pi \times r^2 \times n^{\circ} \text{ de vueltas} \times n^{\circ} \text{ cables} = \pi \times 0.225^2 \times 18 \times 4 = 11.45 \text{ mm}^2$$

$$aws = \pi \times r^2 \times n^{\circ} \text{ de vueltas} \times n^{\circ} \text{ cables} = \pi \times 0.225^2 \times 4 \times 14 \times 2 = 17.81 \text{ mm}^2$$

$$awp + aws = 29.26 \text{ mm}^2$$

6. Comprobar que los bobinados entran en la ventana del transformador comparando la sección de los cables con la ventana, minorada por el factor de apilamiento:

$$awp + aws < aw \text{ máx} = kw \times aw$$

$$29.26 \text{ mm}^2 < 0.4 \times 380 \text{ mm}^2 = 152 \text{ mm}^2$$

Comprobamos que, efectivamente, caben los bobinados en la ventana. Con esto, finalizamos el dimensionamiento del transformador. Se han utilizado un total de 28 metros de cable para construirlo.

### 4.3. Dimensionamiento Bobina de resonancia $L_R$

El principio de funcionamiento es el de un tanque de resonancia entre la bobina y la capacidad equivalente de los MOSFET y el transformador. Se puede calcular dicha capacitancia con la siguiente fórmula. Se estima una capacidad del transformador de 50pF:

$$Ceq = 2 \times Coss \text{ avg} + Ctr = 2 \times 190pF + 50pF = 430 pF$$

La energía que debe eliminarse en la transición se calcula con:

$$Wcr = \frac{1}{2} \times Ceq \times Vbus^2$$

Para garantizar una conmutación suave en la “pierna atrasada” (*lagging-leg*) del convertidor, la energía acumulada en  $L_r$ , debe ser mayor que la almacenada en el condensador equivalente:

$$Wlr = \frac{1}{2} \times Lr \times I^2 > \frac{1}{2} \times Ceq \times Vbus^2 = Wcr$$

Se selecciona  $L_r$  para que la frecuencia de resonancia sea cuatro veces la transición máxima de corriente en el transformador:

$$Ttr\_max = \frac{Tr}{4}$$

En la que el periodo es:

$$Tr = \frac{2\pi}{\omega r}$$

Donde omega, es la frecuencia de resonancia:

$$\omega r = \frac{1}{\sqrt{Ceq \times Lr}}$$

Combinando las fórmulas anteriores, se obtiene:

$$Lr = \frac{1}{\omega r^2 \times Ceq} = \left( \frac{2 \times Tr_{max}}{\pi} \right)^2 \times \frac{1}{Cr}$$

Realmente, podemos aproximar la bobina necesaria calculando una corriente crítica en la que el convertidor entra en modo *hard-switching*. Probamos con una bobina de 33uH:

$$I_{crítica} = V_{bus} \times \sqrt{\frac{Ceq}{Lr}} = 400V \times \sqrt{\frac{430pF}{33uH}} = 1.44 A$$

Lo que nos da un régimen ZVS de:

$$I_p \text{ RMS} = \frac{1000W}{400V} = 2.5A$$

$$\text{Régimen ZVS} = \frac{1.44A}{2.5A} \times 100 = 57.6 \%$$

Por lo que se alcanza una conmutación ZVS al 57.6 % con una bobina de 33uH. Se ha elegido implementar una bobina de este valor para una corriente de 10A.

#### 4.4. Dimensionamiento MOSFET rectificación síncrona

Se calculan las pérdidas estimas de conducción y conmutación:

$$Pérdidas \text{ MOSFET} = I_p^2 \times Rds \text{ (ON)} + 2 \times Qg \times Vg \times \frac{f_{sw}}{2}$$

$$Pérdidas \text{ MOSFET} = 16^2 \times 0.023 \text{ ohmios} + 2 \times 45 \times 10^{-9} C \times 20V \times \frac{100 \text{ KHz}}{2}$$

$$Pérdidas \text{ MOSFET} = 5.98 W$$

Como hay dos MOSFET, que cuentan como fuentes de calor, se deberá calcular el disipador para una potencia de:

$$Pérdidas \text{ MOSFET totales} = 2 \times 5.98 W = 11.96 W$$

#### 4.5. Diseño etapa de filtrado en el secundario

Para dimensionar la bobina debemos establecer una corriente de rizado:

$$I_o \text{ rizado} = \frac{Po \times \% \text{ rizado}}{V_o \text{ min}} = \frac{1000W \times 20\%}{57V} = 3.51 A$$

Calculamos la inductancia mínima que cumple esta condición:

$$L_o \text{ min} = \frac{Vl \text{ max}}{I_o \text{ rizado}} \times T_{on}$$

Calculamos primero la tensión máxima en la bobina y el tiempo de transición:

$$Vl \text{ máx} = \frac{Ns}{Np} \times Vin \text{ max} - Vout \text{ min} = \frac{420V}{4.5} - 57V = 36 V$$

$$Ton = \zeta \times Ts = \frac{1}{2} \times \frac{Vout \text{ min}}{Vin \text{ max}} \times \frac{Np}{Ns} \times Ts = \frac{1}{2} \times \frac{57}{420} \times 5 \times 10^{-5} = 3.39 \text{ us}$$

Por lo que, volviendo a la ecuación anterior:

$$Lo \text{ min} = \frac{Vl \text{ max}}{Io \text{ rizado}} \times Ton = \frac{36 V}{3.51 A} \times 3.39 \times 10^{-6} = 34.76 \text{ uH}$$

Por lo que se implementa una bobina de 33uH y 28A de intensidad máxima.

Para calcular la capacidad necesaria para filtrar el ruido de la conmutación, se utilizan condensadores de baja ESR de la marca Nichicon. Se eligen condensadores de 470uF y 100V de tensión máxima.

Se calcula la corriente transitoria del secundario:

$$I_{transient} = \frac{Pout \times Lo \text{ min} \times 0.9}{Vout \text{ min}^2} = \frac{1000W \times 33uH \times 0.9}{57^2} = 9.14 \times 10^{-6} \text{ segundos}$$

Por lo que la capacidad a la salida será:

$$Cout > \frac{\Delta Ic}{10\% \frac{\Delta Vc}{\Delta t}} = \frac{\frac{1000W \times 0.9}{57V}}{\frac{0.1 \times 3}{9.14 \times 10^{-6}}} = 481 \text{ uF}$$

## 5. Dimensionamiento del sistema de refrigeración de la fuente

### 5.1. Incremento de temperatura en el rectificador y los componentes del convertidor Boost PFC

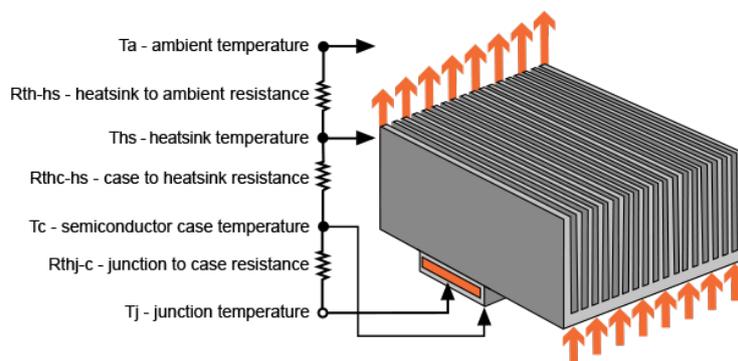


Ilustración 199 - Esquema de consideraciones para el cálculo de disipadores de calor

Primero, determinamos la temperatura máxima de operación de los dispositivos, en este caso, vamos a suponer 100°C para todos ellos. Se supone que la temperatura ambiente

máxima es de 25°C. Por lo que calculamos  $R_{th}$  para la potencia que disipa el conjunto. En este caso, hay que sumar las pérdidas en el puente de diodos, el MOSFET y el diodo del convertidor, lo que nos da un total de 25W:

$$R_{th} = \frac{T_1 - T_2}{P} = \frac{100^{\circ}\text{C} - 25^{\circ}\text{C}}{19.5\text{ W}} = 3.84\text{ }^{\circ}\text{C/W}$$

Se ha instalado un disipador con un coeficiente térmico aproximado de 4°C/W, que, con la refrigeración activa de los ventiladores, se podría considerar un rendimiento según la aproximación:

$$F = \frac{4.32}{\sqrt{D}}$$

Donde D, es el volumen de aire en  $\text{m}^3/\text{h}$ , por lo que sustituyendo con los datos de la hoja de características del ventilador FFB0412VHN:

$$F = \frac{4.32}{\sqrt{0.447 \times 60 \text{ min}}} = 0.83$$

Una vez obtenemos el factor F, podemos estimar el incremento del rendimiento en el disipador al multiplicarlo por este factor:

$$\text{Rendimiento ventilación forzada} = 4^{\circ}\text{C/W} \times 0.83 = 3.32^{\circ}\text{C/W}$$

Lo que nos da lugar a un incremento de temperatura en los dos convertidores que podemos aproximar de la siguiente forma:

$$\text{Etapa PFC} = \text{Pérdidas Etapa PFC} \times 3.32^{\circ}\text{C/W} + 25^{\circ}\text{C}$$

$$\text{Etapa PFC} = 19.5\text{ W} \times 3.32^{\circ}\text{C/W} + 25^{\circ}\text{C} = 49.42^{\circ}\text{C} + 25^{\circ}\text{C} = 89.7^{\circ}\text{C}$$

Obtenemos una temperatura máxima en el disipador de 89.7°C, aceptable para los componentes.

## 5.2. Incremento de temperatura en los MOSFET del convertidor DC – DC

Podemos suponer los mismos cálculos para los MOSFET del convertidor DC – DC que en el apartado anterior:

$$\text{Convertidor Full BRidge} = \text{Pérdidas Conv. FB} \times 3.32^{\circ}\text{C/W} + 25^{\circ}\text{C}$$

$$\text{Convertidor Full BRidge} = 15.09\text{ W} \times 3.32^{\circ}\text{C/W} + 25^{\circ}\text{C} = 74.8^{\circ}\text{C}$$

Lo que nos deja una temperatura de operación de 74.8°C, que cumple con los requerimientos.

## 5.3. Disipador MOSFET rectificación síncrona

En la etapa de rectificación síncrona, se ha instalado un disipador CR101-50, con un rendimiento de 5.3°C/W, lo que nos da un incremento de temperatura de:

$$MOSFET\ SR = Pérdidas\ MOSFET\ SR \times 5.3^{\circ}C/W + 25^{\circ}C$$

$$MOSFET\ SR = 11.96W \times 5.3^{\circ}C/W + 25^{\circ}C = 88.4^{\circ}C$$

La temperatura de los transistores de la rectificación síncrona, dado el flujo de aire de la fuente podemos esperar que sea algo menor, aun así, está dentro de los límites aceptables.

## 5.4. Dimensionamiento MOSFET de control de ventiladores

Pérdidas resultantes en cada transistor:

$$I_{on} = \frac{P_{out}}{n \times V_{in}} = \frac{12\ W}{0,93 \times 12V} = 1,075\ A$$

$$P_{conducción} = I_{on}^2 \times R_{ds} (on) \times \Delta 80\% = 1,075^2 \times 0,028\ \Omega \times 1,8 = 0,058\ W$$

$$P_{conducción\ totales} = P_{conducción} \times 2 = 0,058\ W \times 2 = 0,116\ W$$

$$P_{sw} = \frac{2}{3} \times C_{oss} \times 5 \times V_{on}^2 \times f_{sw} = \frac{2}{3} \times 920\ pF \times 5 \times (12\ V)^2 \times 32\ KHz = 0,015\ W$$

$$P_{sw\ totales} = P_{sw} \times 2 = 0,015 \times 2 = 0,03\ W$$

$$Pérdidas\ totales\ MOSFET = 0,116\ W + 0,015W = 0,15\ W$$

## 6. Cálculos de escala de medidas analógicas

### 6.1. . Lectura de la tensión en el secundario

Para ello, se ha utilizado un divisor de tensión que presenta las siguientes características:

$$V_{out} = V_{inmax} \times \frac{R_2}{R_1 + R_2} = 60V \times \frac{10k}{120k + 10k} = 4.61V$$

Calculamos la sensibilidad del divisor en decenas de milivoltios. Tenemos que encontrar una sensibilidad que nos permita detectar un cambio mínimo en la tensión:

$$Valor\ 1 = 50.00\ V \times \frac{10k}{120k + 10k} = 3.8461V$$

$$Valor\ 2 = 50.02\ V \times \frac{10k}{120k + 10k} = 3.8476V$$

$$Sensibilidad = Valor\ 2 - Valor\ 1 = 3.8476V - 3.8461V = 1.5mV$$

Comparamos la sensibilidad obtenida con la que tiene el ADC del microcontrolador:

$$Sensibilidad\ ADC = \frac{AVDD}{2^{n^{\circ}\ de\ bits}} = \frac{5V}{2^{12\ bits}} = 1.22mV$$

Al ser mayor la sensibilidad del divisor de tensión que la del ADC, quiere decir que nos permite detectar cambios mínimos en la tensión del secundario de 20mV.

## 6.2. Lectura de la corriente en el secundario

Se presentan a continuación, los cálculos del circuito y la sensibilidad de la medida:

$$Ganancia\ operacional = 5 + \frac{80K}{Rg} = 5 + \frac{80K}{4k7} = 22.02$$

$$Valor\ muestra = (V_{shunt} + V_{offset}) \times G$$

$$Valor\ 1 = (1A \times 0.01\ ohmios + 250uV) \times 22.02 = 225.7\ mV$$

$$Valor\ 2 = (1.01A \times 0.01\ ohmios + 250uV) \times 22.02 = 227.9\ mV$$

$$Sensibilidad = Valor\ 2 - Valor\ 1 = 227.9\ mV - 225.7\ mV = 2.2\ mV$$

Como muestran las fórmulas, la sensibilidad en incrementos de 1mA es de 2.2mV, por lo que cumple con la sensibilidad de 1.22mV de nuestro ADC.

El máximo rango de tensión de la lectura es:

$$Valor\ 16A = (16A \times 0.01\ ohmios + 250uV) \times 22.02 = 3.5287\ V$$

Esta tensión cumple con el nivel máximo de salida del amplificador, que es  $VCC - 0.9V$ , ya que no es rail-to-rail.

## 7. Cálculos controlador DC – DC

### 7.1. Escalado lecturas analógicas

Para el control del convertidor, debemos ser capaces de leer tres variables:

- Corriente en el primario.
- Tensión en el secundario.
- Corriente en el secundario.

El conversor A/D del microcontrolador, es de 12 bits, y nuestra referencia de tensión es de 5V. Lo que quiere decir que cuando lleguemos a dicha tensión, el conversor nos dará una lectura de 4095 (0xFFFh).

Para adaptar la lectura en el primario, conocemos que la salida máxima está en 4V, que equivale a 3.72A, por lo que establecemos un límite en el programa cuando la lectura pase de:

$$Límite\ primario = \frac{4V}{5V} \times 4096 = 3277\ (0xCCDh)$$

$$Escala\ I.\ Prim = \frac{4V}{5V} = 0.8$$

$$Lectura Escala Completa = \frac{3.72V}{0.8} = 4.65 A$$

Por lo que, si obtenemos 5V en la entrada, leeremos 4.65A.

Conocemos que la salida del operacional que lee la tensión del secundario es, como máximo 4.61V, por lo que debemos escalar la medida para que coincida con 60V:

$$Escala V.Sec = \frac{4.61V}{5V} = 0.922$$

$$Lectura Escala Completa = \frac{60V}{0.922} = 65.08 V$$

Por lo que, si obtenemos 5V en la entrada, leeremos 65V.

Para adaptar la corriente en el secundario, debemos hacer la misma escala que para la tensión:

$$Escala I.Sec = \frac{3.528V}{5V} = 0.7056$$

$$Lectura Escala Completa = \frac{16A}{0.7056} = 22.676 A$$

Por lo que, si obtenemos 5V en la entrada, tendremos 22.676A en la salida.

## 7.2. Cálculo del controlador 2P2Z

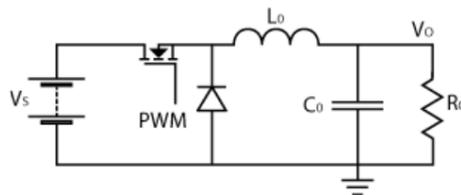


Ilustración 200 - Se puede considerar el convertidor como un reductor básico

Podemos modelar nuestro convertidor como un reductor (Buck) que presenta la siguiente ecuación de transferencia bajo un sistema de control Tipo II de pico de corriente:

$$Hc(s) = \left(\frac{\omega_{CP0}}{s}\right) \times \frac{\left(1 + \frac{s}{\omega_{CZ1}}\right)}{\left(1 + \frac{s}{\omega_{CP1}}\right)}$$

Donde el polo,  $\omega_{CP1}$ , se debe ajustar para cancelar el efecto de la resistencia equivalente de la salida ESR, que presenta la siguiente fórmula:

$$\omega_{CP1} = \frac{1}{R_{ESR} \times C_0}$$

El cero,  $\omega_{CZ1}$ , se coloca para obtener un margen de fase adecuado, y, por último, el polo,  $\omega_{CP0}$ , se debe colocar para obtener una frecuencia de cruce deseada,  $f_x$ . Generalmente colocándolo al 20% de esta frecuencia se obtiene un margen de fase adecuado.

$$\omega_{CZ1} = \frac{1}{5} \times 2\pi \times f_x$$

$$\omega_{CP0} = \frac{1.23 f_x R_i (L_o + 0.32 R_o T_s) R_1 R_2}{L_o R_o}$$

Donde:

$$R_1 = \sqrt{1 - 4f_x^2 T_s^2 + 16f_x^4 T_s^4}$$

$$R_2 = \sqrt{1 + \frac{39.48 C_o^2 f_x^2 L_o^2 R_o^2}{(L_o + 0.32 R_o T_s)^2}}$$

Se hacen estos cálculos en radianes/s no en Hz.

Podemos estimar, a partir de una frecuencia de cruce de 10 KHz, los valores de estas ecuaciones, teniendo en cuenta una ESR de 0.1 ohmios, una bobina de 33uH y una capacidad de 1410 uF. Realmente, estos cálculos, deberán ser ajustados al sistema real cuando se ponga en funcionamiento.

Los datos aquí expuestos sirven de ejemplo, pero no son los valores definitivos para ser implementados:

$$\omega_{CP1} = 73314 \text{ rads}^{-1} \text{ (11668 Hz)}$$

$$\omega_{CZ1} = 12566 \text{ rads}^{-1} \text{ (2000 Hz)}$$

$$\omega_{CP0} = 162464 \text{ rads}^{-1} \text{ (25857 Hz)}$$

Sustituyendo los valores obtenidos en la ecuación anterior:

$$H_c(s) = \left( \frac{162464}{s} \right) \times \frac{\left( 1 + \frac{s}{12566} \right)}{\left( 1 + \frac{s}{73314} \right)}$$

Se ha escrito esta ecuación en MATLAB y, mediante la aproximación bilinear de Tustin con un tiempo de muestro de 16.66 microsegundos, obtenemos la función en tiempo discreto con los coeficientes que debemos implementar:

```

Controlador_z =

    5.415 z^2 + 1.026 z - 4.389
-----
    z^2 - 1.242 z + 0.2417
    
```

*Ilustración 201 - Controlador implementado en el programa*

Además, podemos obtener el diagrama de Bode del controlador para comprobar que es estable y que tenemos un adecuado margen de fase:

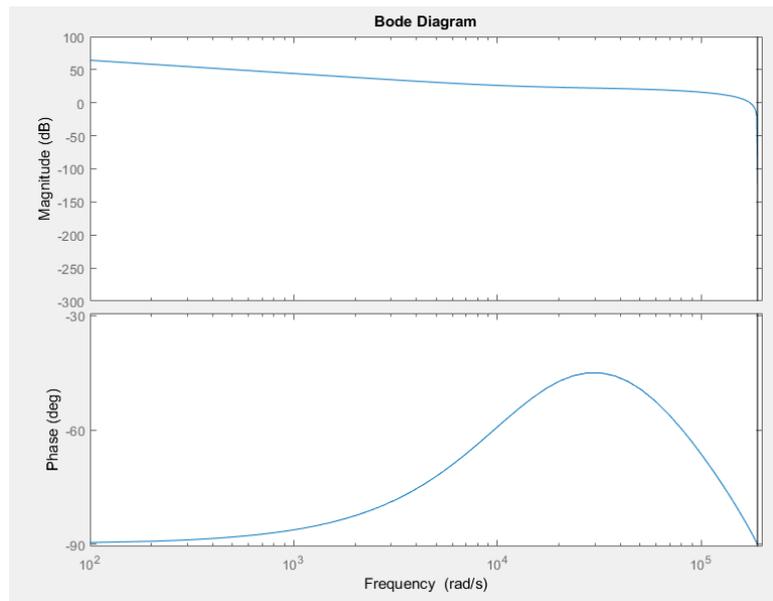


Ilustración 202 - Diagrama de Bode del controlador con una  $F_s = 60000$  Hz, presenta estabilidad y adecuado margen de fase

### 7.3. Cálculo de la rampa de compensación digital

Para eliminar el polo de alta frecuencia de la función de transferencia localizado en la mitad de la frecuencia de muestreo, se utiliza una rampa moduladora que atenúa este efecto. Presenta la siguiente fórmula:

$$V_{pp} = - \frac{(0.18 - D) \times R_i \times T_s \times V_{in}}{L_o}$$

En la que podemos calcular:

$$D = \frac{V_{in} \text{ (Entrada Convertidor)}}{V_{out} \text{ (Varía con la consigna)}} = \frac{400V}{20V \text{ (Por ejemplo)}}$$

$$R_i = \frac{V_{out} \text{ máxima Transformador de corriente}}{\text{Corriente Primario}} = \frac{4V}{3.72A}$$

$$T_s = \frac{1}{F_s} = \frac{1}{60000 \text{ Hz}} = 16.66 \text{ us}$$

$$L_o = 33\mu H$$

Esta rampa se calcula cuando se declara el controlador, no es necesario calcularla todo el tiempo.

El DAC que viene después, tiene 15 bits, lo que equivaldría a una tensión de 5V, por lo que podemos adaptar la altura de la rampa a un valor discreto:

$$\text{Rampa} = V_{pp} \times \frac{n^{\circ} \text{ de bits DAC}}{5V}$$

El valor para decrementar cada ciclo de ejecución es entonces:

$$DecVal = \frac{Rampa \times 60000Hz}{60000000 Hz}$$

Firmado, a 31 de agosto de 2019, Logroño



Alberto Otero Olavarrieta



# ANEXO II: SOFTWARE UTILIZADO



## Índice

1. Software de diseño 3D y circuitos impresos .....	155
1.1. FUSION 360 .....	155
1.2. Eagle.....	155
1.3. Integración y trabajo en la nube.....	156
2. Matlab.....	156
3. SATURN PCB Design Inc. PCB Toolki.....	157
4. LabVIEW.....	157
5. Arduino IDE .....	157
6. MPLAB X IDE y MPLAB CODE CONFIGURATOR.....	158
7. Nextion Editor.....	159
8. Cura 3D.....	159

## Ilustraciones

Ilustración 1 - Logotipo Fusion 360 .....	155
Ilustración 2 - Logotipo GRABCAD.....	155
Ilustración 3 - Logotipo EAGLE CAD .....	155
Ilustración 4 – Interfaz de sincronización con Fusion 360 en EAGLE .....	156
Ilustración 5 - Logotipo MATLAB.....	156
Ilustración 6 - Logotipo SATURN PCB Design .....	157
Ilustración 7 - Logotipo LabVIEW .....	157
Ilustración 8 - Interfaz Arduino IDE .....	157
Ilustración 9 - Placa de desarrollo Arduino Nano.....	158
Ilustración 10 - Logotipos MPLAB X IDE y MPLAB Code Configurator .....	158
Ilustración 11 - Programadora PICKit3 .....	158
Ilustración 12 - Interfaz Nextion Editor .....	159
Ilustración 13 - Logotipo Cura 3D.....	159



## 1. Software de diseño 3D y circuitos impresos

Para realizar el diseño de todos los componentes eléctricos y mecánicos, se ha utilizado una combinación de programas CAD (Diseño Asistido por Ordenador) que han facilitado el desarrollo y la realización de los modelos, así como la generación de planos.

Se ha elegido software de Autodesk para esta tarea. Ya que sus programas están disponibles de manera gratuita a los alumnos de la UR bajo una licencia para estudiantes de 3 años de duración.

### 1.1. FUSION 360



*Ilustración 203 - Logotipo FUSION 360*

FUSION 360 es una herramienta de modelado y simulación 3D, que tiene como particularidad, el trabajo en la nube. Esto permite trabajar desde cualquier ordenador y siempre con la versión más reciente del modelo. FUSION 360 también guarda un historial de versiones de nuestras piezas, que podemos revertir en cualquier momento.

FUSION 360 se ha utilizado para realizar los modelos del Panel frontal y Panel trasero, así como las láminas de acero con el módulo especializado para máquinas plegadoras.

También se ha utilizado para verificar todos los montajes de las placas de circuitos impresos como se explicará después. Se han obtenido todos los modelos de transistores, resistencias, bobinas, etc. desde la base de modelos 3D de dominio público GRABCAD.

# GRABCAD

*Ilustración 204 - Logotipo GRABCAD*

### 1.2. Eagle

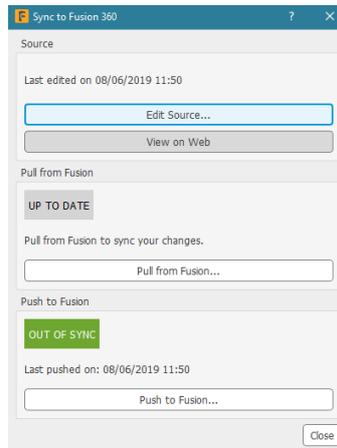


*Ilustración 205 - Logotipo EAGLE*

Para el diseño de circuitos impresos, se ha utilizado EAGLE, una herramienta de diseño profesional, con librerías para gran cantidad de los componentes del mercado.

Realmente, la elección de las herramientas de diseño es subjetiva a cada usuario, pero en este proyecto, se ha optado por esta herramienta gracias a la integración que tiene con FUSION 360. Lo que ha supuesto un gran ahorro de tiempo en generación de modelos 3D y planos.

### 1.3. Integración y trabajo en la nube



*Ilustración 206 – Interfaz de sincronización con FUSION 360 en EAGLE*

El trabajo en la nube de FUSION 360 permite trabajar desde cualquier ordenador al que podamos acceder al programa con nuestra cuenta y usuario. También nos permite subir diseños de otros programas en diversos formatos.

Desde las últimas actualizaciones, EAGLE y FUSION 360, permiten trabajar de manera sincronizada, dando acceso a nuestra placa de circuito impreso en un modelo 3D listo para preparar o simular su montaje.

## 2. Matlab



*Ilustración 207 - Logotipo MATLAB*

Para la simulación del control del convertidor DC - DC se ha implementado un algoritmo de control, cuyos parámetros se han simulado utilizando las herramientas especializadas del software Matlab. Nos permite conocer la respuesta del controlador a varios impulsos, así como su comportamiento en el dominio frecuencial.

### 3. SATURN PCB Design Inc. PCB Toolki



Ilustración 208 - Logotipo SATURN PCB Design

Para comprobar que las placas de circuito impreso, cumplen todas las condiciones de diseño, se ha utilizado esta herramienta de cálculo gratuita. Se ha comprobado que el ancho de las pistas cumple las necesidades de corriente exigidas, así como que la separación entre pistas era suficiente para evitar arcos eléctricos entre ellas.

### 4. LabVIEW



Ilustración 209 - Logotipo LabVIEW

LabVIEW permite una programación rápida de comunicaciones entre dispositivos con las librerías VISA; es esta una de las razones que han llevado a la elección de esta herramienta para la implementación del control bajo PC. También se ha utilizado para crear una interfaz sencilla, para que el usuario pueda mandar consignas a la fuente de alimentación y establecer unos parámetros de carga de baterías.

### 5. Arduino IDE

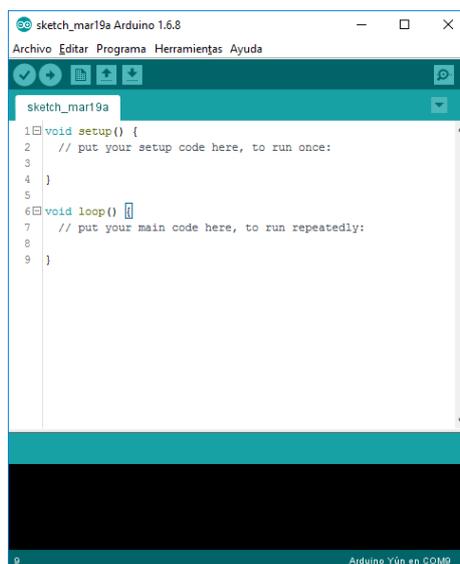


Ilustración 210 - Interfaz Arduino IDE

El entorno de programación C/C++ Arduino IDE, permite una programación rápida de los microcontroladores ATmega328P y ATtiny84, dada su sencillez y la gran cantidad de usuarios que documentan muchas de sus funcionalidades. Permite utilizar una placa Arduino UNO convencional como programador ISP para estos procesadores.



*Ilustración 211 - Placa de desarrollo Arduino Nano*

## 6. MPLAB X IDE y MPLAB CODE CONFIGURATOR



*Ilustración 212 - Logotipos MPLAB X IDE y MPLAB Code Configurator*

MPLAB X IDE, es el entorno de programación de Microchip para su gama de procesadores de 8, 16 y 32 bits. Contiene los compiladores necesarios para cada producto de la marca y ofrece una programación C/C++ convencional.

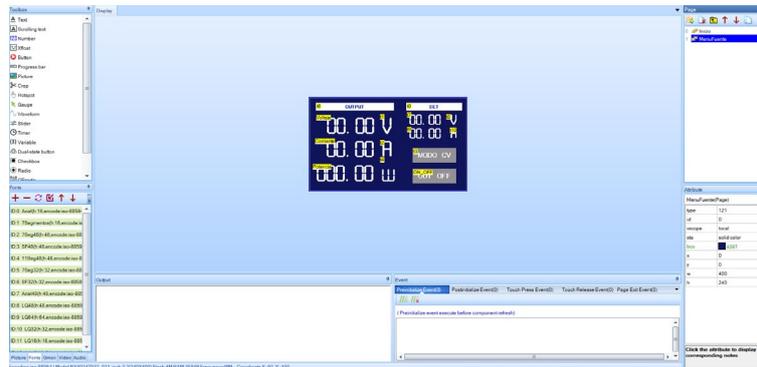
Recientemente, dada la complejidad para configurar ciertas funcionalidades de los procesadores, Microchip desarrolló MPLAB CODE CONFIGURATOR. Este software permite seleccionar de manera gráfica las funciones que queremos activar o utilizar sin tener que pasar por la hoja de datos del producto en cuestión. Así, en poco tiempo, se pueden tener preparados los periféricos que usamos en el proyecto.

Para programar los procesadores, se ha utilizado una tarjeta programadora PICKit 3.



*Ilustración 213 - Programadora PICKit3*

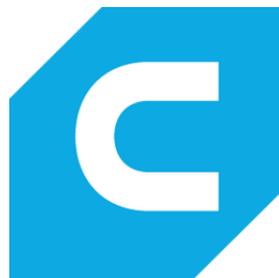
## 7. Nextion Editor



*Ilustración 214 - Interfaz Nextion Editor*

Para la programación de pantallas HMI Nextion, el fabricante proporciona la herramienta Nextion Editor. Consiste en un entorno en que se pueden añadir objetos de texto o imágenes, así como botones y otros actuadores para el control táctil. También permite programar eventos, como temporizaciones.

## 8. Cura 3D



*Ilustración 215 - Logotipo Cura 3D*

La impresión de piezas en 3D, requiere de un programa capaz de separar el modelo en capas, que puedan simular su impresión, así como convertir el modelo a g-code de la impresora específica utilizada para la impresión. Estos programas reciben el nombre de *slicers*. Se han generado los archivos STL con Fusion 360, y se ha decidido utilizar el software gratuito Cura 3D para su impresión.



# ANEXO III: PRUEBAS



## Índice

1. Instrumentos utilizados .....	165
1.1. BitScope.....	165
1.2. Generador de señales FeelTech FY3200 .....	165
1.3. ANENG AN8008 .....	166
1.4. Hantek dso5102P .....	166
2. Pruebas.....	167
2.1. Entorno de programación.....	167
2.2. Prototipo de pruebas fuente completa .....	167
2.3. Pruebas realizadas en filtro EMI .....	168
2.4. Pruebas realizadas en convertidor PFC .....	169
2.5. Pruebas realizadas en control de temperatura.....	171
2.6. Pruebas realizadas en driver de puente completo.....	171
2.7. Pruebas realizadas en driver de rectificación síncrona.....	171
2.8. Pruebas realizadas en placa de botones y encoders .....	171
2.9. Pruebas realizadas en placa de gestión del panel frontal .....	172
2.10. Pruebas realizadas en placa de controlador principal.....	172

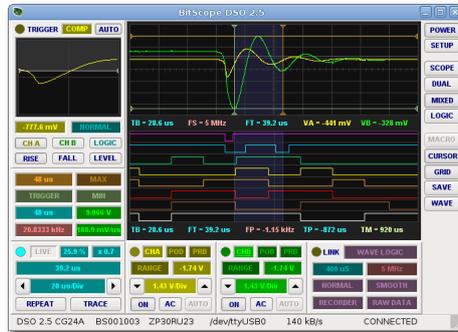
## Ilustraciones

Ilustración 1 - Entorno de trabajo BitScope. Fuente: BitScope.com.....	165
Ilustración 2 - Dispositivo BitScope. Fuente: BitScope.....	165
Ilustración 3 - Generador de señales FY3200. Fuente: Aliexpress .....	165
Ilustración 4 - Multímetro AN800. Fuente: Aliexpress.....	166
Ilustración 5 - Osciloscopio Hantek 5102P 100 MHz .....	166
Ilustración 6 - Entorno de programación del controlador y comunicaciones USB .....	167
Ilustración 7 -Prototipo de la fuente completa.....	168
Ilustración 8 - Prototipo de la fuente completa.....	168
Ilustración 9 - Prototipo original del filtro EMI (Arriba), segunda versión (Abajo) .....	168
Ilustración 10 – Modificaciones realizados en convertidor PFC .....	169
Ilustración 11 - Daños en el MOSFET del convertidor PFC. Se observa grieta en el cuerpo del transistor .....	170
Ilustración 12 -MOSFET dañado del convertidor PFC fuera de la placa. Los daños recibidos provocan que se fragmente en dos mitades.....	170
Ilustración 13 - Modificaciones realizados en placa de botones y encoders.....	171



## 1. Instrumentos utilizados

### 1.1. BitScope



*Ilustración 216 - Entorno de trabajo BitScope. Fuente: BitScope.com*

BitScope es un osciloscopio, generador de señales y analizador lógico USB que se ha utilizado para probar las señales PWM del controlador, ya que son cuatro y en un osciloscopio tradicional solo hay dos canales. También se ha utilizado para generar señales de prueba para testear algunas de las funciones de las placas.



*Ilustración 217 - Dispositivo BitScope. Fuente: BitScope*

### 1.2. Generador de señales FeelTech FY3200



*Ilustración 218 - Generador de señales FY3200. Fuente: Aliexpress*

Se trata de un generador de señales de 20 MHz que se ha utilizado para comprobar los drivers de puente completo y rectificación síncrona. Al tener dos canales, se pueden generar dos señales complementarias, útiles para esta aplicación.

### 1.3. ANENG AN8008



*Ilustración 219 - Multímetro AN800. Fuente: Aliexpress*

Se trata de un multímetro de 4 dígitos (9999 cuentas) que se ha utilizado para todas las lecturas de tensión y corriente de todas las placas.

### 1.4. Hantek dso5102P



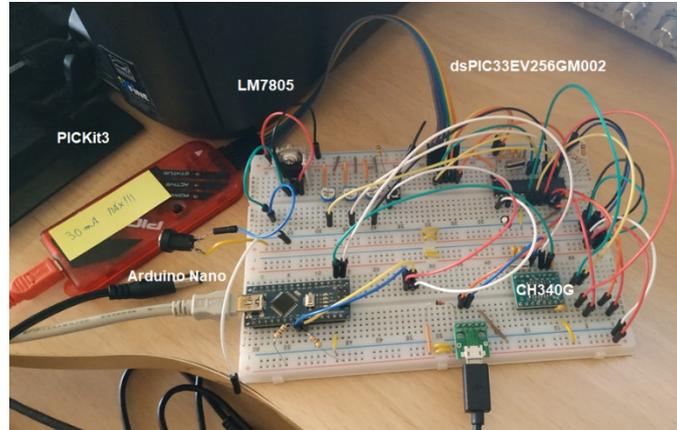
*Ilustración 220 - Osciloscopio Hantek 5102P 100 MHz*

El osciloscopio Hantek 5102P tiene un ancho de banda de 100 MHz y dos canales, así como la posibilidad de guardar capturas de pantalla en una memoria USB. Se trata de un osciloscopio tradicional que se ha utilizado, sobre todo, en las lecturas necesarias para la programación del puerto I2C y USB del controlador principal y el control bajo PC.

Se ha utilizado también para tomar diversas capturas de las señales de los drivers y del control de temperatura.

## 2. Pruebas

### 2.1. Entorno de programación



*Ilustración 221 - Entorno de programación del controlador y comunicaciones USB*

Para programar las distintas funciones del controlador, el control bajo PC y gestión de botones, así como el control de temperatura, se ha utilizado una placa de prototipos diseñada para estas tareas.

En la imagen superior, se puede observar que permite probar las comunicaciones entre los dos controladores principales y las comunicaciones USB.

### 2.2. Prototipo de pruebas fuente completa

La fuente de alimentación se ha montado de forma completa para comprobar que todas las medidas mecánicas coinciden y que se puede terminar su montaje de principio a fin.

También se han realizado las pruebas eléctricas del convertidor PFC con el prototipo montado, ya que se considera más seguro al disponer de un anclaje para la toma de tierra y un conector AC en el panel trasero.



Ilustración 222 -Prototipo de la fuente completa

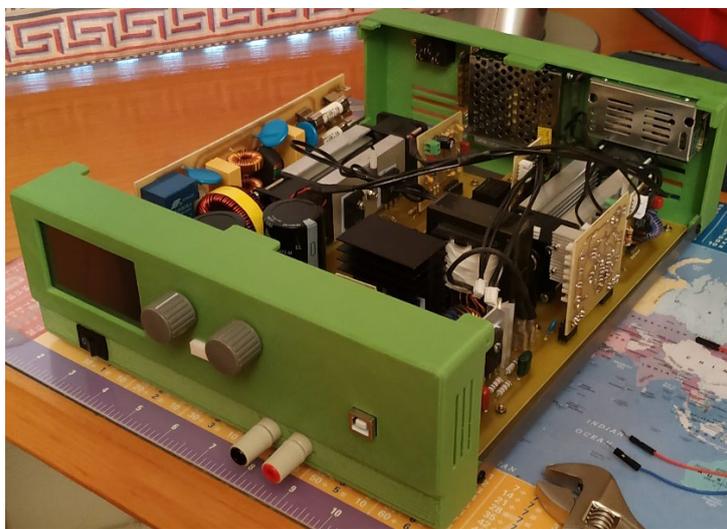


Ilustración 223 - Prototipo de la fuente completa

### 2.3. Pruebas realizadas en filtro EMI



Ilustración 224 - Prototipo original del filtro EMI (Arriba), segunda versión (Abajo)

Para probar el filtro EMI, se utiliza una señal de alterna de 12VAC generada por un transformador, para comprobar que se comunica la entrada con la salida y que no hay

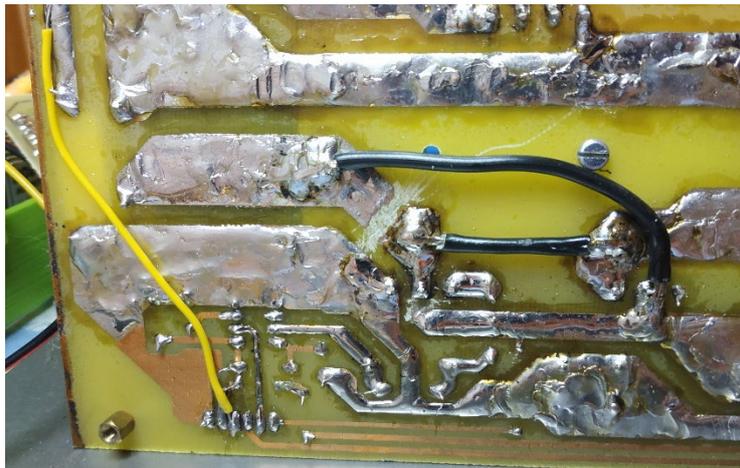
ningún cortocircuito. Se comprueba con el multímetro que efectivamente la placa funciona.

Dadas las grandes dimensiones iniciales de la placa, se producían interferencias mecánicas en el montaje, así como balanceos y vibraciones. Por esta razón, se decide fabricar un segundo prototipo más pequeño y con un montaje vertical más ajustado y firme al convertidor.

## 2.4. Pruebas realizadas en convertidor PFC

Las primeras pruebas realizadas en el convertidor PFC, se realizan con una toma de 12VAC de un transformador para comprobar que se obtiene la tensión rectificada en la salida y que no hay ningún cortocircuito en la placa.

Acto seguido, se conecta la placa del controlador PFC y se comprueba que el convertidor no funciona, no es capaz de elevar la tensión lo suficiente. Se examinan las pistas y se encuentran fallos en el cableado de las mismas. Dada la potencia y corriente que deben soportar estas pistas, se utilizan cables de 10A para unir las pistas correctamente. También se comprueba que la lectura del shunt estaba desconectada.



*Ilustración 225 – Modificaciones realizados en convertidor PFC*

En una segunda prueba, se conecta a la red de 230VAC el convertidor, con el filtro EMI y el controlador PFC. En esta prueba se activa la protección diferencial del cuadro eléctrico poco antes de alcanzar el convertidor la tensión de línea de 400VDC, lo que nos indica que hay corrientes de fuga a tierra.

Se decide cambiar la bobina de 2 mH original del filtro EMI, por una de 5mH, la que se considera que estaba calculada para una frecuencia de corte (35 KHz), que proporcionaba muy poca atenuación. La nueva frecuencia de corte se establece en 25 KHz. El convertidor esta vez alcanza la tensión de línea de 400 VDC y se produce el arranque suave correctamente.



*Ilustración 226 - Daños en el MOSFET del convertidor PFC. Se observa grieta en el cuerpo del transistor*

En una tercera prueba, se conecta por error, el multímetro en modo de lectura de corriente a los terminales de salida del convertidor. Esto produce daños en los siguientes componentes:

- Resistencias de arranque suave en cortocircuito.
- Rectificador KU1510 en cortocircuito.
- Driver IR2125 sin señal en la salida.
- MOSFET IRFP460 en cortocircuito.



*Ilustración 227 -MOSFET dañado del convertidor PFC fuera de la placa. Los daños recibidos provocan que se fragmente en dos mitades*

Al no disponer de recambios para todos los componentes, no se realizan más pruebas de funcionamiento en el convertidor PFC.

De haberse completado la reparación, se hubieran realizado pruebas de funcionamiento con diversas cargas, en las que se hubiera comprobado que, efectivamente, se obtiene un factor de potencia unidad.

## 2.5. Pruebas realizadas en control de temperatura

Para las pruebas del control de temperatura, se realiza la programación del microcontrolador ATtiny84 con un Arduino Nano en una placa de prototipos y se monta en la tarjeta. Se comprueba que se leen todas las tensiones correctas y que no hay cortocircuitos en la placa.

Se realizan pruebas de simulación de la lectura NTC con potenciómetros en los divisores de tensión, se comprueba que se controla la velocidad de los ventiladores y se toman capturas con el osciloscopio

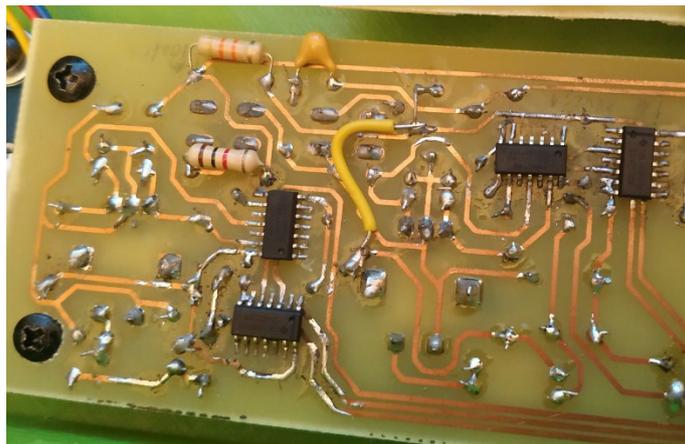
## 2.6. Pruebas realizadas en driver de puente completo

Se conectan a una alimentación de 12V los dos drivers de puente completo, y con unas señales de 100 KHz complementarias al 50% de ciclo de trabajo, se comprueba que funcionan tanto el driver MCP1404, como el transformador P0584NL.

## 2.7. Pruebas realizadas en driver de rectificación síncrona

Se conecta a una alimentación de 12V, y con unas señales de 100 KHz complementarias al 50% de ciclo de trabajo, se comprueba que funciona correctamente el driver MCP1404.

## 2.8. Pruebas realizadas en placa de botones y encoders



*Ilustración 228 - Modificaciones realizados en placa de botones y encoders*

En la placa de antirrebote y montaje de botones y encoders, se realizaron distintas pruebas y medidas con el osciloscopio, para comprobar que todos los botones generaban las señales correspondientes. Se encontraron diversos fallos en el cableado

y disposición de algunos elementos que impedían su montaje, por lo que se realizaron modificaciones para su correcto funcionamiento.

## 2.9. Pruebas realizadas en placa de gestión del panel frontal

Se conecta a una alimentación de 12V y se prueba la tensión del carril de 5V del regulador de tensión, y se verifica que no hay cortocircuitos en la placa. Se programa el microcontrolador y se comprueban las conexiones con la placa de botones y que se leen los valores correctamente. También se prueban las comunicaciones con el puerto USB.

## 2.10. Pruebas realizadas en placa de controlador principal

Se conecta a una alimentación de 12V y se prueban las tensiones más relevantes: bus de 5V del regulador de tensión, referencia de tensión y bus negativo del TC7660. Se comprueba que no hay cortocircuitos en la placa, se programa el microcontrolador mediante la interfaz para el PICkit 3 y se prueba que funciona correctamente.

# ANEXO IV: HOJAS DE DATOS



## Índice

1. Filtro EMI.....	177
2. Controlador PFC.....	177
3. Convertidor PFC.....	177
4. Convertidor DC – DC.....	178
5. Drivers Puente completo.....	179
6. Driver rectificación síncrona.....	180
7. Controlador de temperatura.....	180
8. Controlador del convertidor DC – DC.....	181
9. PCB Botones y encoders.....	181
10. Controlador de gestión de botones y USB.....	182
11. Etapa de salida.....	183
12. Otros.....	183



## 1. Filtro EMI

- CHIN. Fusible RT14 38 mm 10A

[http://www.mantech.co.za/Datasheets/Products/RT28-FUSE\\_CHINT.pdf](http://www.mantech.co.za/Datasheets/Products/RT28-FUSE_CHINT.pdf)

- Songle. SLA12VDC-SL-A

[http://images.100y.com.tw/pdf\\_file/29-SLA-12VDC-SL-C.pdf](http://images.100y.com.tw/pdf_file/29-SLA-12VDC-SL-C.pdf)

- Bourns. 20d431k

<https://www.bourns.com/data/global/pdfs/MOV20D.pdf>

- MMT. Pin Header 1x6. 2.54 mm

<http://www.mmtmachrone.com/pdf/20090217224253PHa.pdf>

- TE Connectivity. 6.3 mm Faston male terminal

<http://www.farnell.com/datasheets/1658358.pdf>

- TENTA. MKP X2 Class. 1.2 uF 275VAC.

<https://mikroshop.ch/pdf/MEX-X2.pdf>

- HVCAP. Y1 Class 4N7 400 VDC.

<https://hv-caps.com/uploads/products/HVC-Safety-Caps-y1.pdf>

## 2. Controlador PFC

- Infineon technologies. ICE3PCS01G

[https://www.infineon.com/dgdl/Infineon-ICE3PCS01-DS-v03\\_00-EN.pdf?fileId=db3a304329a0f6ee0129a67ae8c02b46](https://www.infineon.com/dgdl/Infineon-ICE3PCS01-DS-v03_00-EN.pdf?fileId=db3a304329a0f6ee0129a67ae8c02b46)

- Phoenix Contact. MPT 0,5/ 2-2,54

[https://www.mouser.es/datasheet/2/324/ItemDetail\\_1725656-920552.pdf](https://www.mouser.es/datasheet/2/324/ItemDetail_1725656-920552.pdf)

- MMT. Pin Header 1x6. 2.54 mm 90 degree.

<http://www.mmtmachrone.com/pdf/20090217224253PHa.pdf>

## 3. Convertidor PFC

- Delta. FFB0412VHN

<https://www.mouser.es/datasheet/2/632/FFB40x40x28mm-515835.pdf>

- EPCOS. B43504 470uF 450V 105°C.

<https://www.components-mart.com/datasheets/d1/B43504B5277M62.pdf>

- Fairchild. BC547

<https://www.mouser.es/datasheet/2/308/BC547-1118654.pdf>

- International Rectifier. IR2125

[https://www.infineon.com/dgdl/Infineon-IR2125-DS-v01\\_00-EN.pdf?fileId=5546d462533600a4015355c85ba31694](https://www.infineon.com/dgdl/Infineon-IR2125-DS-v01_00-EN.pdf?fileId=5546d462533600a4015355c85ba31694)

- VISHAY. IRFP460

<http://www.vishay.com/docs/91237/91237.pdf>

- ONSEMI. MUR3060PT

<https://www.onsemi.com/pub/Collateral/MUR3020PT-D.PDF>

- SEP Electronic Corp. KBU1510

[https://img.ozdisan.com/ETicaret\\_Dosya/471087\\_4174433.pdf](https://img.ozdisan.com/ETicaret_Dosya/471087_4174433.pdf)

- Type DME. Polyester Film Capacitors 630V

<https://www.mouser.es/datasheet/2/88/DME-23637.pdf>

- TE Connectivity. 6.3 mm Faston male terminal

<http://www.farnell.com/datasheets/1658358.pdf>

- HVCAP. Y1 Class 4N7 400 VDC.

<https://hv-caps.com/uploads/products/HVC-Safety-Caps-y1.pdf>

#### 4. Convertidor DC – DC

- VISHAY IRFP460

<http://www.vishay.com/docs/91237/91237.pdf>

- TDK. EE55/28/21

[https://www.tdk-electronics.tdk.com/inf/80/db/fer/e\\_55\\_28\\_21.pdf](https://www.tdk-electronics.tdk.com/inf/80/db/fer/e_55_28_21.pdf)

- TDK. PC40

<https://www.theferritegateway.com/uploads/datasheets/TKD/PC40.pdf>

- Delta. FFB0412VHN

<https://www.mouser.es/datasheet/2/632/FFB40x40x28mm-515835.pdf>

- OHMITE. CR101-50

[https://www.ohmite.com/assets/docs/sink\\_cr.pdf?r=false](https://www.ohmite.com/assets/docs/sink_cr.pdf?r=false)

- International Rectifier. IRFP90N20D

<https://xdevs.com/doc/xDevs.com/Neutron/pdf/irfp90n20d.pdf>

- Type DME. Polyester Film Capacitors 250V  
<https://www.mouser.es/datasheet/2/88/DME-23637.pdf>
- Type DME. Polyester Film Capacitors 630V  
<https://www.mouser.es/datasheet/2/88/DME-23637.pdf>
- Panasonic. Aluminum electrolytic capacitors.  
<https://docs-apac.rs-online.com/webdocs/154d/0900766b8154d05e.pdf>
- PULSE Electronics. PE63587NL  
<https://www.mouser.es/datasheet/2/336/P578-1526780.pdf>
- TE Connectivity. 6.3 mm Faston male terminal  
<http://www.farnell.com/datasheets/1658358.pdf>
- ONSEMI. MUR3060PT  
<https://www.onsemi.com/pub/Collateral/MUR3020PT-D.PDF>
- Stackpole Electronics. MPR Series Flameproof Metal Plate Resistor 0.01R 5W  
<https://www.seielect.com/Catalog/SEI-mpr.pdf>
- HVCAP. Y1 Class 4N7 400 VDC.  
<https://hv-caps.com/uploads/products/HVC-Safety-Caps-y1.pdf>

## 5. Drivers Puente completo

- Microchip. MCP1404  
<http://ww1.microchip.com/downloads/en/devicedoc/22022b.pdf>
- PULSE Electronics. P0584NL  
<https://www.mouser.es/datasheet/2/336/-369999.pdf>
- VISHAY. 1N4148  
<https://www.vishay.com/docs/81857/1n4148.pdf>
- MMT. Pin Header 1x4. 2.54 mm  
<http://www.mmtmachrone.com/pdf/20090217224253PHa.pdf>
- XH Connector. 4B-XH-A  
<http://www.farnell.com/datasheets/5448.pdf>
- Type DME. Polyester Film Capacitors 250V  
<https://www.mouser.es/datasheet/2/88/DME-23637.pdf>

## 6. Driver rectificación síncrona

- Microchip. MCP1404

<http://ww1.microchip.com/downloads/en/devicedoc/22022b.pdf>

- VISHAY. 1N4148

<https://www.vishay.com/docs/81857/1n4148.pdf>

- MMT. Pin Header 1x3. 2.54 mm

<http://www.mmtmachrone.com/pdf/20090217224253PHa.pdf>

- XH Connector. 4B-XH-A

<http://www.farnell.com/datasheets/5448.pdf>

- Type DME. Polyester Film Capacitors 250V

<https://www.mouser.es/datasheet/2/88/DME-23637.pdf>

## 7. Controlador de temperatura

- MURATA. NTC 10k B3950

<https://www.murata.com/~media/webrenewal/support/library/catalog/products/thermistor/ntc/r44e.ashx>

- Atmel. ATtiny84

[http://ww1.microchip.com/downloads/en/devicedoc/Atmel-7701\\_Automotive-Microcontrollers-ATtiny24-44-84\\_Datasheet.pdf](http://ww1.microchip.com/downloads/en/devicedoc/Atmel-7701_Automotive-Microcontrollers-ATtiny24-44-84_Datasheet.pdf)

- XH Connector. 2B-XH-A

<http://www.farnell.com/datasheets/5448.pdf>

- VISHAY. IRFZ44

<https://www.vishay.com/docs/91291/91291.pdf>

- MMT. Pin Header 2x5. 2.54 mm

<http://www.mmtmachrone.com/pdf/20090217224253PHa.pdf>

- Phoenix Contact. MPT 0,5/ 2-2,54

<https://www.sparkfun.com/datasheets/Prototyping/Connectors/4UCON%20screw%20terminaI%2020086.pdf>

- STMicroelectronics. LM78L05

<https://www.st.com/resource/en/datasheet/l78l.pdf>

## 8. Controlador del convertidor DC – DC

- XH Connector. 2B-XH-A

<http://www.farnell.com/datasheets/5448.pdf>

- XH Connector. 4B-XH-A

<http://www.farnell.com/datasheets/5448.pdf>

- MMT. Pin Header 1x6. 2.54 mm

<http://www.mmtmachrone.com/pdf/20090217224253PHa.pdf>

- MMT. Pin Header 2x5. 2.54 mm

<http://www.mmtmachrone.com/pdf/20090217224253PHa.pdf>

- Texas Instruments. INA126P

<http://www.ti.com/lit/ds/symlink/ina126.pdf>

- Microchip. MCP6V26

<http://ww1.microchip.com/downloads/en/devicedoc/25007b.pdf>

- Microchip. dsPIC33EV256GM002

<http://ww1.microchip.com/downloads/en/DeviceDoc/70005144e.pdf>

- Microchip. TC7660

<http://ww1.microchip.com/downloads/en/devicedoc/21465c.pdf>

- Advanced Monolithic Systems. AMS1117-5.0

<http://www.advanced-monolithic.com/pdf/ds1117.pdf>

- Fairchild. TL431A

[https://cdn.sparkfun.com/assets/learn\\_tutorials/4/2/3/TL431.pdf](https://cdn.sparkfun.com/assets/learn_tutorials/4/2/3/TL431.pdf)

- ONSEMI. BC337

<https://www.onsemi.com/pub/Collateral/BC337-D.PDF>

- IQD. HC49/4H

<https://www.mouser.es/datasheet/2/741/LFXTAL036183Bulk-998468.pdf>

## 9. PCB Botones y encoders

- Pro-signal. ABI-009-RC. Buzzer

<http://www.farnell.com/datasheets/2171929.pdf>

- ONSEMI. BC337

<https://www.onsemi.com/pub/Collateral/BC337-D.PDF>

- Tyco Electronics. 787834-1 USB Type B

[https://www.te.com/commerce/DocumentDelivery/DDEController?Action=showdoc&DocId=Data+Sheet%7F7-1773442-0%7F0306%7Fpdf%7FEnglish%7FENG\\_DS\\_7-1773442-0\\_0306.pdf%7F292336-1](https://www.te.com/commerce/DocumentDelivery/DDEController?Action=showdoc&DocId=Data+Sheet%7F7-1773442-0%7F0306%7Fpdf%7FEnglish%7FENG_DS_7-1773442-0_0306.pdf%7F292336-1)

- HDK Tactile Switches.

[https://components101.com/sites/default/files/component\\_datasheet/Push-Button.pdf](https://components101.com/sites/default/files/component_datasheet/Push-Button.pdf)

- ALPS Electric Co. Rotary Encoders

[http://www.farnell.com/datasheets/1702859.pdf?\\_ga=2.239202432.2015755544.1566883785-985524540.1508852298](http://www.farnell.com/datasheets/1702859.pdf?_ga=2.239202432.2015755544.1566883785-985524540.1508852298)

- Nexperia. 74HC14

[https://assets.nexperia.com/documents/data-sheet/74HC\\_HCT14.pdf](https://assets.nexperia.com/documents/data-sheet/74HC_HCT14.pdf)

- Texas Instruments. CD4013

<http://www.ti.com/lit/ds/symlink/cd4013b.pdf>

- MMT. Pin Header 1x7. 2.54 mm

<http://www.mmtmachrone.com/pdf/20090217224253PHa.pdf>

- MMT. Pin Header 1x8. 2.54 mm

<http://www.mmtmachrone.com/pdf/20090217224253PHa.pdf>

## 10. Controlador de gestión de botones y USB

- Atmel. ATmega328P

[http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P\\_Datasheet.pdf](http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf)

- Texas Instruments. LM7805

<https://www.sparkfun.com/datasheets/Components/LM7805.pdf>

- XH Connector. 2B-XH-A

<http://www.farnell.com/datasheets/5448.pdf>

- MMT. Pin Header 1x7. 2.54 mm

<http://www.mmtmachrone.com/pdf/20090217224253PHa.pdf>

- MMT. Pin Header 1x8. 2.54 mm

<http://www.mmtmachrone.com/pdf/20090217224253PHa.pdf>

- WCH. CH340G

<https://cdn.sparkfun.com/datasheets/Dev/Arduino/Other/CH340DS1.PDF>

- IQD. HC49/4H

<https://www.mouser.es/datasheet/2/741/LFXTAL036183Bulk-998468.pdf>

- IQD. HC49/4H

<https://www.mouser.es/datasheet/2/741/LFXTAL036183Bulk-998468.pdf>

## 11. Etapa de salida

- TE Connectivity. 6.3 mm Faston male terminal

<http://www.farnell.com/datasheets/1658358.pdf>

- XH Connector. 4B-XH-A

<http://www.farnell.com/datasheets/5448.pdf>

## 12. Otros

- RS PRO. Black SPST Rocker Switch. 16A 125A 10A 250VAC

<https://docs-emea.rs-online.com/webdocs/1582/0900766b81582c7f.pdf>

- Bulgin. IEC C14

<http://www.farnell.com/datasheets/311707.pdf>

- Nextion. NX4024K032

<https://nextion.tech/datasheets/nx4024k032/>

- FTDI. FT232

[https://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS\\_FT232R.pdf](https://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232R.pdf)

- Microchip. MCP2200

<https://ww1.microchip.com/downloads/en/DeviceDoc/22228A.pdf>

- Analog Devices. AD8230

<https://www.analog.com/media/en/technical-documentation/data-sheets/AD8230.pdf>

- Analog Devices. AD8422

<https://www.analog.com/media/en/technical-documentation/data-sheets/AD8422.pdf>

- Analog Devices. ADP3605

<https://www.analog.com/media/en/technical-documentation/data-sheets/ADP3605.pdf>

- MeanWell. SCWN06A-12

<http://www.meanwellusa.com/productPdf.aspx?i=858#1>

- MORNSUN. CUWB1212YMD-6WR3

[https://www.mornsun-power.com/uploads/pdf/CUWB\\_YMD-6WR3.pdf](https://www.mornsun-power.com/uploads/pdf/CUWB_YMD-6WR3.pdf)

# ANEXO V: CÓDIGO



## Índice

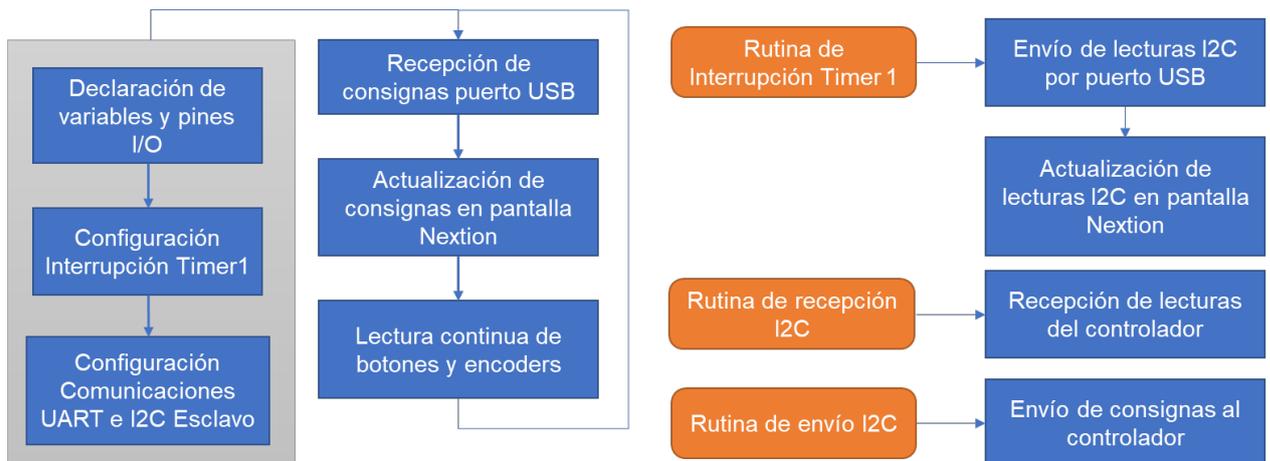
1.	Programa Arduino Pantalla_Botones_USB_I2C .....	189
1.1.	Diagrama de flujo .....	189
1.2.	Código.....	189
2.	Programa Control de temperatura .....	200
2.1.	Diagrama de flujo .....	200
2.2.	Código.....	201
3.	Programa controlador DC – DC .....	204
3.1.	Diagrama de flujo .....	204
3.2.	Código.....	205
3.2.1.	clock.c.....	205
3.2.2.	adc1.c.....	206
3.2.3.	mcc.c .....	208
3.2.4.	I2C.....	212
3.2.5.	pin_manager.c .....	224
3.2.6.	pwm.c .....	227
3.2.7.	reset.c.....	231
3.2.8.	System.c .....	234
3.2.9.	tmr1.c.....	235
3.2.10.	traps.c.....	241
3.2.11.	Variables.h.....	243
3.2.12.	Variables.c .....	245
3.2.13.	main.c.....	247
4.	Programa Control remoto fuente de alimentación LabVIEW.....	252
4.1.	Interfaz de usuario .....	252
4.2.	Variables locales.....	253
4.3.	Diagrama de bloques .....	254
4.4.	Bucles .....	254
4.5.	Dependencias.....	263
4.6.	Jerarquía.....	264
4.7.	Programa LabVIEW – Prueba comunicaciones.vi.....	265
4.8.	Interfaz de usuario .....	265
4.9.	Variables locales.....	266
4.10.	Diagrama de bloques.....	266

4.11.	Bucles .....	267
4.12.	Dependencias .....	268
4.13.	Jerarquía .....	268
4.14.	Programa LabVIEW – Formar Cadena.vi .....	269
4.15.	Interfaz de usuario .....	269
4.16.	Variables locales .....	269
4.17.	Diagrama de bloques.....	269
4.18.	Bucles .....	270
4.19.	Jerarquía .....	270

## 1. Programa Arduino Pantalla\_Botones\_USB\_I2C

A continuación, se adjunta el código encargado de leer los botones del panel frontal y de gestionar las cadenas para la comunicación USB a través de rutinas de interrupción. Así como de enviar las consignas que escribe el usuario al microcontrolador principal, a través del puerto I2C.

### 1.1. Diagrama de flujo



### 1.2. Código

```

// ----- // Librerías utilizadas // ----- //

#include <Nextion.h>
#include <Wire.h>
#include <NeoSWSerial.h>

// ----- // PINES I/O // ----- //

int BOTON_OUT =7;
int BOTON_FN = 6;
int buzzer = 2;
int BOTON_ENCV = 15;
int BOTON_ENCC = 4;
int STEPV = 17;
int DIRV = 16;
int STEPC = 5;
int DIRC = 14;

// ----- // VARIABLES // ----- //

int ByteRecibido;
int SalidaONOFF;
int OUT = 0;
int CCCV = 0;
int Temp;
  
```

```
int CambioOUT = 0;
int CambioCCCV = 0;
int cnt = 0;
int count;
int FINE;
float Voltaje = 5.00;
float Corriente = 4.00;
float Potencia = 0.00;
String VoltajeI2C;
String CorrienteI2C;
float CalVoltaje = 0;
float CalCorriente = 0;
float IncVolt = 1;
float IncCorr = 1;
int NuevoVoltaje = 0;
int NuevaCorriente = 0;
int ModoRemoto = 0;
int Conexion = 0;
int CambioModoRemoto = 0;
int ActivarLV = 0;
int tempLabView = 0;
int tempOUT = 0;
int len = 10;
char ConsignaI2C[16];
char LecturaI2C[15];

String CadenaI2C;
String CadenaLecturaI2C;
String TempI2C;
String VConsigna = "05.00";
String CConsigna = "05.00";
String Recepcion;
String ConsignaLV;
String DatosParaLV;
String V, C, P;
String SVoltaje, SCorriente, SCCC, SOUT;

int EstEncV, LastEstEncV, EstEncC, LastEstEncC;

NeoSWSerial nextion(9,8); // RX, TX

Nextion Pantalla(nextion, 9600);

// ----- //

void setup() {

// ----- INTERRUPTO TIMER 1 ----- //

cli(); //stop interrupts

//set timer1 interrupt at 5Hz (200ms)
TCCR1A = 0;// set entire TCCR1A register to 0
TCCR1B = 0;// same for TCCR1B
TCNT1 = 0;//initialize counter value to 0

// set compare match register for 1hz increments
OCR1A = 3124;// = (16*10^6) / (5*1024) - 1 (must be <65536)

// turn on CTC mode
TCCR1B |= (1 << WGM12);
```

```

// Set CS10 and CS12 bits for 1024 prescaler
TCCR1B |= (1 << CS12) | (1 << CS10);

// enable timer compare interrupt
TIMSK1 |= (1 << OCIE1A);

sei(); //allow interrupts

// ----- //

pinMode(BOTON_ENCV, INPUT);
pinMode(BOTON_ENCC, INPUT);
pinMode(BOTON_FN, INPUT);
pinMode(BOTON_OUT, INPUT);
pinMode(STEPV, INPUT);
pinMode(STEPC, INPUT);
pinMode(DIRV, INPUT);
pinMode(DIRC, INPUT);

Wire.begin(1); // join i2c bus with address #8
Wire.onReceive(receiveEvent); // register event
Wire.onRequest(requestEvent); // register event

Serial.begin(38400);
nextion.begin(38400);
Pantalla.init();

// Precarga Cadena I2C
CadenaI2C = VConsigna + ',' + CConsigna + ',' + String(OUT);
CadenaI2C.toCharArray(ConsignaI2C, 14);
}

// ----- // RUTINA DE INTERRUPTON TIMER 1 // -----
// ----- //

ISR(TIMER1_COMPA_vect) {

// ----- // COMUNICACIÓN LABVIEW // -----
//

if (ActivarLV == 1 && OUT == 1)
{
VoltajeI2C = CadenaLecturaI2C.substring(0,5);
CorrienteI2C = CadenaLecturaI2C.substring(6,11);
SCCCV = CadenaLecturaI2C.substring(12,13);
TempI2C = CadenaLecturaI2C.substring(14,15);

Temp= TempI2C.toInt();
CalVoltaje = VoltajeI2C.toFloat();
CalCorriente = CorrienteI2C.toFloat();

if (CalVoltaje < 10)
V = '0' + String(CalVoltaje);
else
V = String(CalVoltaje);
if (CalCorriente < 10)
C = '0' + String(CalCorriente);
else
C = String(CalCorriente);
}
}

```

```
Potencia = (CalVoltaje*CalCorriente);

if (Potencia < 10)
    P = "00" + String(Potencia);
else if (Potencia >= 10 && Potencia < 100)
    P = "0" + String(Potencia);
else
    P = String(Potencia);

    DatosParaLV = V + ',' + C + ',' + P + ',' + String(OUT) +
',,' + SCCC + ',' + TempI2C; // 5 + 1 + 5 + 1 + 6 + 1 + 1 + 1 + 1 + 1
+ 1 = 24 bytes
    Serial.println(DatosParaLV);
}

if (ActivarLV == 1 && OUT == 0)
{
    DatosParaLV = String("00.00,00.00,000.00") + ',' +
String(OUT) + ',' + SCCC + ',' + TempI2C;
    Serial.println(DatosParaLV);
}

// ----- // COMUNICACIÓN PANTALLA // -----
//

VoltajeI2C = CadenaLecturaI2C.substring(0,5);
CorrienteI2C = CadenaLecturaI2C.substring(6,11);
SCCC = CadenaLecturaI2C.substring(12,13);
TempI2C = CadenaLecturaI2C.substring(14,15);

Temp= TempI2C.toInt();
CalVoltaje = VoltajeI2C.toFloat();
CalCorriente = CorrienteI2C.toFloat();

if (OUT == 1)
{
    if (CalVoltaje < 10)
        Pantalla.setComponentText("Voltaje", "0" +
String(CalVoltaje, 2));
    else
        Pantalla.setComponentText("Voltaje", String(CalVoltaje, 2));

    if (CalCorriente < 10)
        Pantalla.setComponentText("Corriente", "0" +
String(CalCorriente, 2));
    else
        Pantalla.setComponentText("Corriente", String(CalCorriente,
2));

    Potencia = CalVoltaje*CalCorriente;

    if (Potencia < 10)
        Pantalla.setComponentText("Potencia", "0" + String(Potencia,
2));
    else if (Potencia >= 10 && Potencia < 100)
        Pantalla.setComponentText("Potencia", String(Potencia, 2));
    else
        Pantalla.setComponentText("Potencia", String(Potencia, 2));
}
```

```
}

void loop() {

// ----- // RECEPCIÓN CONSIGNAS LABVIEW // -----
// ----- //

    if (Serial.available() > 0) {

        Recepcion = Serial.readString();
        Recepcion.trim(); // Recorta el
caracter de terminación de la cadena para poder compararlo

        if (Recepcion == "LabViewON")
        {
            ActivarLV = 1;
            ModoRemoto = 1;
            CambioModoRemoto = 1;
            Recepcion = ""; // Limpia la
cadena para otra recepcion
            Conexion = 1;
            Pantalla.setComponentText("Voltaje", "00.00");
// Resetea los valores en pantalla
            Pantalla.setComponentText("Corriente", "00.00");
            Pantalla.setComponentText("Potencia", "00.00");
        }

        else if (Recepcion == "LabViewOFF")
        {
            ActivarLV = 0;
            CambioModoRemoto = 1;
            OUT = 0;
            CCCV = 0;
            CambioOUT = 1;
            CambioCCCV = 1;
            ModoRemoto = 0;
            Conexion = 0;
            Recepcion = ""; // Limpia la
cadena para otra recepcion

            Pantalla.setComponentText("Voltaje", "00.00");
// Resetea los valores en pantalla
            Pantalla.setComponentText("Corriente", "00.00");
            Pantalla.setComponentText("Potencia", "00.00");
        }

        if (Conexion == 1)
        {

            ConsignaLV = Recepcion;

            SVoltaje = ConsignaLV.substring(0, 4);
            SCorriente = ConsignaLV.substring(6, 10);
            SOUT = ConsignaLV.substring(12);

            Voltaje = SVoltaje.toFloat();
            Corriente = SCorriente.toFloat();
            OUT = SOUT.toInt();
            CambioOUT = 1;
        }
    }
}
```

```
        // Aqui se monta la cadena para I2C y se actualiza el
buffer. Solo se actualiza si hay datos en el puerto serie que vienen
de LabView

        if (Voltaje < 10)
            VConsigna = '0' + String(Voltaje);
        else
            VConsigna = String(Voltaje);

        if (Corriente < 10)
            CConsigna = '0' + String(Corriente);
        else
            CConsigna = String(Corriente);

        CadenaI2C = VConsigna + ',' + CConsigna + ',' + SOUT;

        CadenaI2C.toCharArray(ConsignaI2C, 14);

    }
}

// ----- // Envia Valores por pantalla // -----
----- //

if (NuevoVoltaje == 1)
{
    if (Voltaje < 10)
        Pantalla.setComponentText("t7", '0'+ String(Voltaje, 2));
    else
        Pantalla.setComponentText("t7", String(Voltaje, 2));

    NuevoVoltaje = 0;
}
if (NuevaCorriente == 1)
{
    if (Corriente < 10)
        Pantalla.setComponentText("t9", '0' + String(Corriente, 2));
    else
        Pantalla.setComponentText("t9", String(Corriente, 2));
    NuevaCorriente = 0;
}

if (CambioOUT == 1)
{
    if (OUT == 1)
    {
        Pantalla.setComponentText("ON_OFF", "OUT ON");

        // Se monta la cadena para el I2C y se envia cada 200ms. Solo
se manda si hay cambio en la salida
        if (Voltaje < 10)
            VConsigna = '0' + String(Voltaje);
        else
            VConsigna = String(Voltaje);

        if (Corriente < 10)
            CConsigna = '0' + String(Corriente);
        else
            CConsigna = String(Corriente);
    }
}
```

```
String(OUT);          CadenaI2C = VConsigna + ',' + CConsigna + ',' +

                      CadenaI2C.toCharArray(ConsignaI2C, 14);

                      }
                      else
                      {
                          Pantalla.setComponentText("ON_OFF", "OUT OFF");
                          Pantalla.setComponentText("Voltaje", "00.00");          // Resetea
los valores en pantalla
                          Pantalla.setComponentText("Corriente", "00.00");
                          Pantalla.setComponentText("Potencia", "00.00");

                          // Se monta la cadena para el I2C y se envia cada 200ms. Solo
se manda si hay cambio en la salida
                          if (Voltaje < 10)
                              VConsigna = '0' + String(Voltaje);
                          else
                              VConsigna = String(Voltaje);

                          if (Corriente < 10)
                              CConsigna = '0' + String(Corriente);
                          else
                              CConsigna = String(Corriente);

                          CadenaI2C = VConsigna + ',' + CConsigna + ',' +

String(OUT);

                          CadenaI2C.toCharArray(ConsignaI2C, 14);

                      }
                      CambioOUT = 0;
                      }

                      if (CambioModoRemoto == 1)
                      {
                          if (ModoRemoto == 1)
                          {
                              Pantalla.setComponentText("t7", "RmOTE");
                              Pantalla.setComponentText("t9", "RmOTE");
                              tone(buzzer, 440, 50);
                              delay(100);
                              tone(buzzer, 440, 50);
                          }
                          else
                          {
                              Voltaje = 0.00;
                              Corriente = 0.00;
                              Pantalla.setComponentText("t7", '0' + String(Voltaje, 2));
                              Pantalla.setComponentText("t9", '0' + String(Corriente, 2));
                              tone(buzzer, 294, 50);
                              delay(100);
                              tone(buzzer, 294, 50);
                          }

                          CambioModoRemoto = 0;
                      }

                      if (Temp == 1)
                      {
```

```
        OUT = 0;
        Pantalla.setComponentText("Voltaje", "TEmp");
        Pantalla.setComponentText("Corriente", "TEmp");
        tone(buzzer, 440, 50);
        delay(100);
        tone(buzzer, 440, 50);
    }

// ----- // Leer Botones // -----
----- //

    if (digitalRead(BOTON_FN) == LOW && ModoRemoto == 0) {

        if (FINE <= 3)
        {
            FINE = FINE + 1;
        }
        if (FINE > 2)
        {
            FINE = 0;
        }

        switch(FINE)
        {
            case 0:
                IncVolt = 1;
                IncCorr = 1;
                break;
            case 1:
                IncVolt = 0.1;
                IncCorr = 0.1;
                break;
            case 2:
                IncVolt = 0.01;
                IncCorr = 0.01;
                break;
        }

        tone(buzzer, 440, 50);
        delay(100);
    }

    if (digitalRead(BOTON_OUT) == LOW && ModoRemoto == 0) {

        tone(buzzer, 440, 50);
        OUT = !OUT;
        CambioOUT = 1;
        delay(200);
    }

    if (digitalRead(BOTON_ENCV) == HIGH && ModoRemoto == 0) {

        if (Voltaje < 10)
            VConsigna = '0' + String(Voltaje);
        else
            VConsigna = String(Voltaje);

        if (Corriente < 10)
            CConsigna = '0' + String(Corriente);
        else
```

```

        CConsigna = String(Corriente);

        CadenaI2C = VConsigna + ',' + CConsigna + ',' + String(OUT);

        CadenaI2C.toCharArray(ConsignaI2C, 14);

        tone(buzzer, 440, 50);
        delay(200);
    }
    if (digitalRead(BOTON_ENCC) == HIGH && ModoRemoto == 0) {

        if (Voltaje < 10)
            VConsigna = '0' + String(Voltaje);
        else
            VConsigna = String(Voltaje);

        if (Corriente < 10)
            CConsigna = '0' + String(Corriente);
        else
            CConsigna = String(Corriente);

        CadenaI2C = VConsigna + ',' + CConsigna + ',' + String(OUT);

        CadenaI2C.toCharArray(ConsignaI2C, 14);

        tone(buzzer, 440, 50);
        delay(200);
    }

// ----- // LECTURA ENCODERS // -----
// ----- //

// Encoder de Voltaje //

EstEncV = digitalRead(STEPV);

if (EstEncV != LastEstEncV) {

    if (EstEncV == HIGH && digitalRead(DIRV) == LOW && ModoRemoto ==
0){
        tone(buzzer, 300, 10);
        if (Voltaje > 0.00)
        {
            Voltaje = Voltaje - IncVolt;
            NuevoVoltaje = 1;

            if (Voltaje < 0.00)
            {
                Voltaje = 0.00;
            }
        }
    }

    if (EstEncV == HIGH && digitalRead(DIRV) == HIGH && ModoRemoto
== 0){
        tone(buzzer, 300, 10);
        if (Voltaje < 60.00)
        {

```

```
        Voltaje = Voltaje + IncVolt;
        NuevoVoltaje = 1;

        if (Voltaje > 60.00)
        {
            Voltaje = 60.00;
        }
    }
}

LastEstEncV = EstEncV;

// Encoder de Corriente //

EstEncC = digitalRead(STEPC);

if (EstEncC != LastEstEncC) {

0) {
    if (EstEncC == HIGH && digitalRead(DIRC) == LOW && ModoRemoto ==
    tone(buzzer, 300, 10);
    if (Corriente > 0.00)
    {
        Corriente = Corriente - IncCorr;
        NuevaCorriente = 1;

        if (Corriente < 0.00)
        {
            Corriente = 0.00;
        }
    }
}

    if (EstEncC == HIGH && digitalRead(DIRC) == HIGH && ModoRemoto
== 0) {
        tone(buzzer, 300, 10);
        if (Corriente < 16.00)
        {
            Corriente = Corriente + IncCorr;
            NuevaCorriente = 1;

            if (Corriente > 16.00)
            {
                Corriente = 16.00;
            }
        }
    }
}

    LastEstEncC = EstEncC;
}

// ----- // FUNCIONES INTERRUPCIÓN I2C // -----
// ----- //

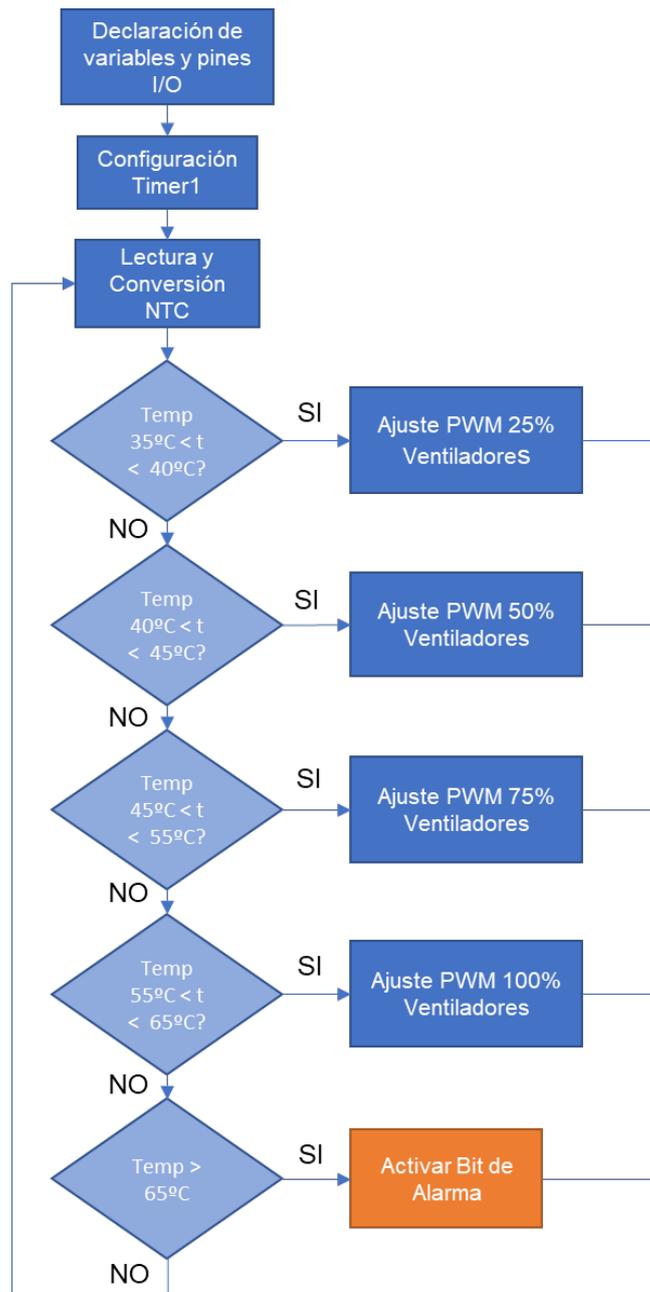
void receiveEvent(int howMany) {
    int count=0;
    while (Wire.available() > 0) { // loop through all but the last
        LecturaI2C[count] = Wire.read();
        count++;
    }
}
```

```
    }  
    CadenaLecturaI2C = String(LecturaI2C);  
}  
  
void requestEvent() {  
    //Wire.write("015.62,05.34,1"); // Cadena de prueba - Responde con  
15 bytes: VoltajeConsigna;CorrienteConsigna;Salida  
    Wire.write(ConsignaI2C);  
  
}
```

## 2. Programa Control de temperatura

El programa aquí descrito, es el encargado de controlar la temperatura de los distintos semiconductores de potencia en los dos convertidores de la fuente. Se trata de un programa que es capaz de leer la temperatura través de un termistor NTC y de variar la velocidad de los ventiladores a través de técnicas PWM

### 2.1. Diagrama de flujo



## 2.2. Código

```

#include <math.h>

// ----- // VARIABLES // -----
// ----- //

int VreadNTC1 = 0;
int VreadNTC2 = 1;
int Vent1 = 5;
int Vent2 = 6;
int Alarma = 8;

float VNTC1;
float VNTC2;
float T0inv;
float Binv;
float T_inversaNTC1;
float T_inversaNTC2;
float adcMax;
float lnADCNTC1;
float lnADCNTC2;
int TempActualNTC1;
int TempActualNTC2;

// ----- //
// ----- //

void setup() {

// ----- // DELARACIÓN I/O // -----
// ----- //

pinMode(Vent1, OUTPUT);
pinMode(Vent2, OUTPUT);
pinMode(VreadNTC1, INPUT);
pinMode(VreadNTC2, INPUT);
pinMode(Alarma, OUTPUT);

// ----- // CONFIGURACIÓN PWM // -----
// ----- //

// WGM10,
WGM12: Fast PWM, 8-bit TOP=255 // CS10: Sin
// COM1A1: Pin
prescaler // y nivel alto
    TCCR1A = _BV(COM1A1) | _BV(COM1B1) | _BV(WGM10); // 32 kHz con
6 a nivel BAJO cuando coincide con OCR1A // 8MHz de reloj
// y nivel alto
con TOP // 32 kHz con
    TCCR1B = _BV(CS10) | _BV(WGM12); // 8MHz de reloj
}

void loop() {

// ----- // LECTURA Y CONVERSIÓN VALORES
NTC // ----- //
    
```

```
VNTC1 = analogRead(VreadNTC1);
VNTC2 = analogRead(VreadNTC2);

T0inv = 1.00/298.15;
Binv = 1.00/3950.00;
adcMax = 1023.00;
lnADCNTC1 = (log((adcMax/VNTC1)-1));
lnADCNTC2 = (log((adcMax/VNTC2)-1));

T_inversaNTC1 = T0inv + Binv * lnADCNTC1;
T_inversaNTC2 = T0inv + Binv * lnADCNTC2;
TempActualNTC1 = ((1/T_inversaNTC1)-273);
TempActualNTC2 = ((1/T_inversaNTC2)-273);

// ----- // CONTROL PRIMER PAR DE
VENTILADORES // ----- //

    if(TempActualNTC1 < 35 ) {
        OCR1A = 0;
    }
    else if(TempActualNTC1 >= 35 & TempActualNTC1 < 40) {
        OCR1A = 50;
    }
    else if(TempActualNTC1 >= 40 & TempActualNTC1 < 45) {
        OCR1A = 125;
    }
    else if(TempActualNTC1 >= 50 & TempActualNTC1 < 55) {
        OCR1A = 200;
    }
    else if(TempActualNTC1 >= 55 & TempActualNTC1 < 65) {
        OCR1A = 255;
    }
    else if (TempActualNTC1 >= 65) {
        OCR1A = 255;
        digitalWrite(Alarma, HIGH);
        delay(2000);
        digitalWrite(Alarma, LOW);
    }

// ----- // CONTROL SEGUNDO PAR DE
VENTILADORES // ----- //

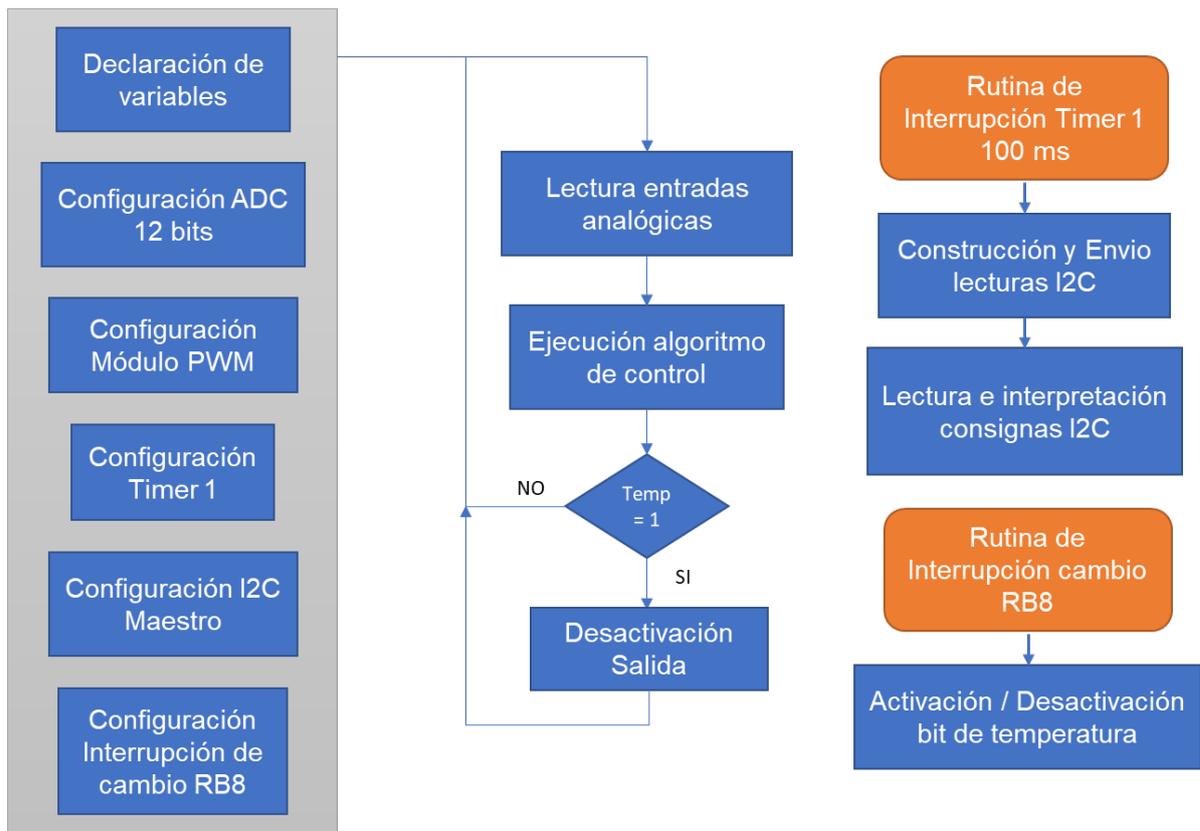
    if(TempActualNTC2 < 35 ) {
        OCR1B = 0;
    }
    else if(TempActualNTC2 >= 35 & TempActualNTC2 < 40) {
        OCR1B = 50;
    }
    else if(TempActualNTC2 >= 40 & TempActualNTC2 < 45) {
        OCR1B = 125;
    }
    else if(TempActualNTC2 >= 45 & TempActualNTC2 < 55) {
        OCR1B = 200;
    }
    else if(TempActualNTC2 >= 55 & TempActualNTC2 < 65) {
        OCR1B = 255;
    }
    else if (TempActualNTC2 >= 65) {
        OCR1B = 255;
        digitalWrite(Alarma, HIGH);
    }
```

```
    delay(2000);  
    digitalWrite(Alarma, LOW);  
  }  
}
```

### 3. Programa controlador DC – DC

Se presenta aquí, el código encargado de la configuración de los periféricos del microcontrolador dsPIC33EV256GM002. Así como el programa principal y las rutinas de interrupción encargadas de controlar la consigna de tensión y corriente de la fuente de alimentación, activar la salida y gestionar las cadenas de la comunicación I2C.

#### 3.1. Diagrama de flujo



## 3.2. Código

### 3.2.1. clock.c

```
#include <stdint.h>
#include "xc.h"
#include "clock.h"

void CLOCK_Initialize(void)
{
    //Temporarily switch to FRC (without PLL), so we can safely change
    the PLL settings,
    //in case we had previously been already running from the PLL.
    __builtin_write_OSCCONH((uint8_t) (OSCCONH & 0xF8));
    if(OSCCONbits.COSC != OSCCONbits.NOSC)
    {
        __builtin_write_OSCCONL((uint8_t) (OSCCONL & 0xFF));
    }
    //Initiate clock switching operation.
    while(OSCCONbits.OSWEN != 0);    //Wait for switching
    complete.
    while (OSCCONbits.LOCK != 1);    //Wait till PLL is locked
    }
    // FRCDIV FRC/2; PLLPRE 2; DOZE 1:8; PLLPOST 1:2; DOZEN disabled;
    ROI disabled;
    CLKDIV = 0x3100;
    // TUN Center frequency;
    OSCTUN = 0x00;
    // ROON disabled; ROSEL FOSC; RODIV 0; ROSSLP disabled;
    REFOCON = 0x00;
    // PLLDIV 58;
    PLLFBD = 0x3A;
    // AD1MD enabled; T3MD enabled; T4MD enabled; T1MD enabled; U2MD
    enabled; T2MD enabled; U1MD enabled; SPI2MD enabled; SPI1MD enabled;
    PWMMD enabled; T5MD enabled; I2C1MD enabled;
    PMD1 = 0x00;
    // IC4MD enabled; IC3MD enabled; OC1MD enabled; IC2MD enabled;
    OC2MD enabled; IC1MD enabled; OC3MD enabled; OC4MD enabled;
    PMD2 = 0x00;
    // CMPMD enabled;
    PMD3 = 0x00;
    // CTMUMD enabled; REFOMD enabled;
    PMD4 = 0x00;
    // PWM2MD enabled; PWM1MD enabled; PWM3MD enabled;
    PMD6 = 0x00;
    // DMA0MD enabled;
    PMD7 = 0x00;
    // DMTMD enabled; SENT2MD enabled; SENT1MD enabled;
    PMD8 = 0x00;
    // CF no clock failure; NOSC PRIPLL; CLKLOCK unlocked; OSWEN
    Switch is Complete;
    __builtin_write_OSCCONH((uint8_t) (0x03));
    __builtin_write_OSCCONL((uint8_t) (0x01));
    // Wait for Clock switch to occur
    while (OSCCONbits.OSWEN != 0);
    while (OSCCONbits.LOCK != 1);
}
```

### 3.2.2. adc1.c

```
#include "adcl.h"

/**
 * Section: Data Type Definitions
 */

/* ADC Driver Hardware Instance Object

@Summary
  Defines the object required for the maintenance of the hardware
  instance.

@Description
  This defines the object required for the maintenance of the
  hardware
  instance. This object exists once per hardware instance of the
  peripheral.

 */
typedef struct
{
    uint8_t intSample;
}

ADC_OBJECT;

static ADC_OBJECT adc1_obj;

/**
 * Section: Driver Interface
 */

void ADC1_Initialize (void)
{
    // ASAM enabled; ADDMABM disabled; ADSIDL disabled; DONE disabled;
    SIMSAM Sequential; FORM Absolute decimal result, unsigned, right-
    justified; SAMP disabled; SSRC Clearing sample bit ends sampling and
    starts conversion; AD12B 12-bit; ADON enabled; SSRCG disabled;

    AD1CON1 = 0x8404;

    // CSCNA disabled; VCFG0 AVDD; VCFG1 AVSS; ALTS disabled; BUFM
    disabled; SMPI Generates interrupt after completion of every
    sample/conversion operation; CHPS 1 Channel;

    AD1CON2 = 0x00;

    // SAMC 1; ADRC FOSC/2; ADCS 7;

    AD1CON3 = 0x107;

    // CH0SA OA2/AN0; CH0SB OA2/AN0; CH0NB VREFL; CH0NA VREFL;

    AD1CHS0 = 0x00;

    // CSS26 disabled; CSS25 disabled; CSS24 disabled; CSS27 disabled;

    AD1CSSH = 0x00;
```

```
// CSS2 disabled; CSS1 disabled; CSS0 disabled; CSS5 disabled;
CSS4 disabled; CSS3 disabled;

AD1CSSL = 0x00;

// DMABL Allocates 1 word of buffer to each analog input; ADDMAEN
disabled;

AD1CON4 = 0x00;

// CH123SA2 disabled; CH123SB2 CH1=OA2/AN0,CH2=AN1,CH3=AN2;
CH123NA disabled; CH123NB CH1=VREF-,CH2=VREF-,CH3=VREF-;

AD1CHS123 = 0x00;

adc1_obj.intSample = AD1CON2bits.SMPI;
}

void __attribute__((weak)) ADC1_CallBack(void)
{
    // Add your custom callback code here
}

void ADC1_Tasks ( void )
{
    if(IFS0bits.AD1IF)
    {
        // ADC1 callback function
        ADC1_CallBack();

        // clear the ADC interrupt flag
        IFS0bits.AD1IF = false;
    }
}

/**
    End of File
*/
```

### 3.2.3. mcc.c

```
/**
 *Generated PIC24 / dsPIC33 / PIC32MM MCUs Source File

 *Company:
   Microchip Technology Inc.

 *File Name:
   mcc.c

 *Summary:
   This is the mcc.c file generated using PIC24 / dsPIC33 / PIC32MM
   MCUs

 *Description:
   This header file provides implementations for driver APIs for all
   modules selected in the GUI.
   Generation Information :
     Product Revision   : PIC24 / dsPIC33 / PIC32MM MCUs - 1.95-b-
   SNAPSHOT
     Device             : dsPIC33EV256GM002
   The generated drivers are tested against the following:
     Compiler           : XC16 v1.36
     MPLAB              : MPLAB X v5.10
 */

/*
 * (c) 2016 Microchip Technology Inc. and its subsidiaries. You may
 * use this
 * software and any derivatives exclusively with Microchip products.

 * THIS SOFTWARE IS SUPPLIED BY MICROCHIP "AS IS". NO WARRANTIES,
 * WHETHER
 * EXPRESS, IMPLIED OR STATUTORY, APPLY TO THIS SOFTWARE, INCLUDING
 * ANY IMPLIED
 * WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A
 * PARTICULAR PURPOSE, OR ITS INTERACTION WITH MICROCHIP PRODUCTS,
 * COMBINATION
 * WITH ANY OTHER PRODUCTS, OR USE IN ANY APPLICATION.

 * IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL,
 * PUNITIVE,
 * INCIDENTAL OR CONSEQUENTIAL LOSS, DAMAGE, COST OR EXPENSE OF ANY
 * KIND
 * WHATSOEVER RELATED TO THE SOFTWARE, HOWEVER CAUSED, EVEN IF
 * MICROCHIP HAS
 * BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO
 * THE
 * FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL
 * CLAIMS IN
 * ANY WAY RELATED TO THIS SOFTWARE WILL NOT EXCEED THE AMOUNT OF
 * FEES, IF ANY,
 * THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THIS SOFTWARE.

 * MICROCHIP PROVIDES THIS SOFTWARE CONDITIONALLY UPON YOUR
 * ACCEPTANCE OF THESE
 * TERMS.
 */
```

```
// Configuration bits: selected in the GUI

// FSEC
#pragma config BWRP = OFF //Boot Segment Write-Protect Bit->Boot
Segment may be written
#pragma config BSS = DISABLED //Boot Segment Code-Protect Level
bits->No Protection (other than BWRP)
#pragma config BSS2 = OFF //Boot Segment Control Bit->No Boot
Segment
#pragma config GWRP = OFF //General Segment Write-Protect Bit-
>General Segment may be written
#pragma config GSS = DISABLED //General Segment Code-Protect Level
bits->No Protection (other than GWRP)
#pragma config CWRP = OFF //Configuration Segment Write-Protect
Bit->Configuration Segment may be written
#pragma config CSS = DISABLED //Configuration Segment Code-Protect
Level bits->No Protection (other than CWRP)
#pragma config AIVTDIS = DISABLE //Alternate Interrupt Vector Table
Disable Bit ->Disable Alternate Vector Table

// FBSLIM
#pragma config BSLIM = 8191 //Boot Segment Code Flash Page Address
Limit Bits->8191

// FOSCSEL
#pragma config FNOSC = PRIPLL //Initial oscillator Source Selection
Bits->Primary Oscillator with PLL module (XT + PLL, HS + PLL, EC +
PLL)
#pragma config IESO = OFF //Two Speed Oscillator Start-Up Bit-
>Start up device with user selected oscillator source

// FOSC
#pragma config POSCMD = HS //Primary Oscillator Mode Select Bits-
>HS Crystal Oscillator mode
#pragma config OSCIOFNC = ON //OSC2 Pin I/O Function Enable Bit-
>OSC2 is general purpose digital I/O pin
#pragma config IOL1WAY = ON //Peripheral Pin Select Configuration
Bit->Allow Only One reconfiguration
#pragma config FCKSM = CSECME //Clock Switching Mode Bits->Both
Clock Switching and Fail-safe Clock Monitor are enabled
#pragma config PLLKEN = ON //PLL Lock Enable Bit->Clock switch to
PLL source will wait until the PLL lock signal is valid

// FWDT
#pragma config WDTPOST = PS32768 //Watchdog Timer Postscaler Bits-
>1:32768
#pragma config WDTPRE = PR128 //Watchdog Timer Prescaler Bit->1:128
#pragma config FWDTEN = OFF //Watchdog Timer Enable Bits->WDT and
SWDTEN Disabled
#pragma config WINDIS = OFF //Watchdog Timer Window Enable Bit-
>Watchdog timer in Non-Window Mode
#pragma config WDTWIN = WIN25 //Watchdog Window Select Bits->WDT
Window is 25% of WDT period

// FPOR
#pragma config BOREN0 = ON //Brown Out Reset Detection Bit->BOR is
Enabled

// FICD
#pragma config ICS = PGD1 //ICD Communication Channel Select Bits-
>Communicate on PGEC1 and PGED1
```

```
// FDMTINTVL
#pragma config DMTIVTL = 0 //Lower 16 Bits of 32 Bit DMT Window
Interval->0

// FDMTINTVH
#pragma config DMTIVTH = 0 //Upper 16 Bits of 32 Bit DMT Window
Interval->0

// FDMTCNTL
#pragma config DMCNTL = 0 //Lower 16 Bits of 32 Bit DMT
Instruction Count Time-Out Value->0

// FDMTCNTH
#pragma config DMCNTH = 0 //Upper 16 Bits of 32 Bit DMT
Instruction Count Time-Out Value->0

// FDMT
#pragma config DMTEN = DISABLE //Dead Man Timer Enable Bit->Dead
Man Timer is Disabled and can be enabled by software

// FDEVOPT
#pragma config PWMLOCK = ON //PWM Lock Enable Bit->Certain PWM
registers may only be written after key sequence
#pragma config ALTI2C1 = OFF //Alternate I2C1 Pins Selection Bit-
>I2C1 mapped to SDA1/SCL1 pins

// FALTREG
#pragma config CXTX1 = NONE //Interrupt Priority Level (IPL)
Selection Bits For Alternate Working Register Set 1->Not Assigned
#pragma config CXTX2 = NONE //Interrupt Priority Level (IPL)
Selection Bits For Alternate Working Register Set 2->Not Assigned

#include "mcc.h"
#include "reset.h"
#include "clock.h"

/**
 * Section: Local Variables
 */

/**
 * Section: Function prototypes
 */

/**
 * a private place to store the error code if we run into a severe
 * error
 */

void OSCILLATOR_Initialize(void)
{
    CLOCK_Initialize();
}

uint16_t SYSTEM_GetResetCause(void)
{
    return RCON;
}

void __attribute__((weak)) SYSTEM_ResetCauseHandler(void)
```

```
{  
    RESET_CauseHandler();  
}  
  
void SYSTEM_ResetCauseClearAll()  
{  
    RESET_CauseClearAll();  
}  
/**  
End of File  
*/
```

### 3.2.4. I2C

```

#include "i2c1.h"

/**
 * Section: Data Types
 */

/**
 * I2C Driver Queue Status Type

 @Summary
   Defines the type used for the transaction queue status.

 @Description
   This defines type used to keep track of the queue status.
 */

typedef union
{
  struct
  {
    uint8_t full:1;
    uint8_t empty:1;
    uint8_t reserved:6;
  }s;
  uint8_t status;
} I2C_TR_QUEUE_STATUS;

/**
 * I2C Driver Queue Entry Type

 @Summary
   Defines the object used for an entry in the i2c queue items.

 @Description
   This defines the object in the i2c queue. Each entry is a composed
   of a list of TRBs, the number of the TRBs and the status of the
   currently processed TRB.
 */
typedef struct
{
  uint8_t count; // a count of
  trb's in the trb list
  I2C1_TRANSACTION_REQUEST_BLOCK *ptrb_list; // pointer to the
  trb list
  I2C1_MESSAGE_STATUS *pTrFlag; // set with the
  error of the last trb sent. // if all trb's
  are sent successfully, // then this is
  I2C1_MESSAGE_COMPLETE
} I2C_TR_QUEUE_ENTRY;

/**
 * I2C Master Driver Object Type

 @Summary
   Defines the object that manages the i2c master.

```

```

    @Description
    This defines the object that manages the sending and receiving of
    i2c master transactions.
    */

typedef struct
{
    /* Read/Write Queue */
    I2C_TR_QUEUE_ENTRY    *pTrTail;        // tail of the queue
    I2C_TR_QUEUE_ENTRY    *pTrHead;       // head of the queue
    I2C_TR_QUEUE_STATUS   trStatus;       // status of the last
transaction
    uint8_t                i2cDoneFlag;    // flag to indicate
the current
                                // transaction is done
    uint8_t                i2cErrors;     // keeps track of
errors

} I2C_OBJECT ;

/**
    I2C Master Driver State Enumeration

    @Summary
    Defines the different states of the i2c master.

    @Description
    This defines the different states that the i2c master
    used to process transactions on the i2c bus.
    */

typedef enum
{
    S_MASTER_IDLE,
    S_MASTER_RESTART,
    S_MASTER_SEND_ADDR,
    S_MASTER_SEND_DATA,
    S_MASTER_SEND_STOP,
    S_MASTER_ACK_ADDR,
    S_MASTER_RCV_DATA,
    S_MASTER_RCV_STOP,
    S_MASTER_ACK_RCV_DATA,
    S_MASTER_NOACK_STOP,
    S_MASTER_SEND_ADDR_10BIT_LSB,
    S_MASTER_10BIT_RESTART,

} I2C_MASTER_STATES;

/**
    Section: Macro Definitions
    */

/* defined for I2C1 */

#ifndef I2C1_CONFIG_TR_QUEUE_LENGTH
    #define I2C1_CONFIG_TR_QUEUE_LENGTH 1
#endif
    
```

```

#define I2C1_TRANSMIT_REG                I2C1TRN
    // Defines the transmit register used to send data.
#define I2C1_RECEIVE_REG                I2C1RCV    // Defines
the receive register used to receive data.

// The following control bits are used in the I2C state machine to
manage
// the I2C module and determine next states.
#define I2C1_WRITE_COLLISION_STATUS_BIT    I2C1STATbits.IWCOL
    // Defines the write collision status bit.
#define I2C1_ACKNOWLEDGE_STATUS_BIT        I2C1STATbits.ACKSTAT
    // I2C ACK status bit.

#define I2C1_START_CONDITION_ENABLE_BIT    I2C1CONLbits.SEN
    // I2C START control bit.
#define I2C1_REPEAT_START_CONDITION_ENABLE_BIT I2C1CONLbits.RSEN //
I2C Repeated START control bit.
#define I2C1_RECEIVE_ENABLE_BIT            I2C1CONLbits.RCEN //
I2C Receive enable control bit.
#define I2C1_STOP_CONDITION_ENABLE_BIT      I2C1CONLbits.PEN
    // I2C STOP control bit.
#define I2C1_ACKNOWLEDGE_ENABLE_BIT         I2C1CONLbits.ACKEN
    // I2C ACK start control bit.
#define I2C1_ACKNOWLEDGE_DATA_BIT          I2C1CONLbits.ACKDT
    // I2C ACK data control bit.

/**
    Section: Local Functions
*/

static void I2C1_FunctionComplete(void);
static void I2C1_Stop(I2C1_MESSAGE_STATUS completion_code);

/**
    Section: Local Variables
*/

static I2C_TR_QUEUE_ENTRY
i2c1_tr_queue[I2C1_CONFIG_TR_QUEUE_LENGTH];
static I2C_OBJECT                i2c1_object;
static I2C_MASTER_STATES          i2c1_state = S_MASTER_IDLE;
static uint8_t                    i2c1_trb_count;

static I2C1_TRANSACTION_REQUEST_BLOCK *p_i2c1_trb_current;
static I2C_TR_QUEUE_ENTRY          *p_i2c1_current = NULL;

/**
    Section: Driver Interface
*/

void I2C1_Initialize(void)
{

    i2c1_object.pTrHead = i2c1_tr_queue;
    i2c1_object.pTrTail = i2c1_tr_queue;
    i2c1_object.trStatus.s.empty = true;
    i2c1_object.trStatus.s.full = false;

    i2c1_object.i2cErrors = 0;

```

```
// initialize the hardware
// Baud Rate Generator Value: I2CBRG 298;
I2C1BRG = 0x12A;
// ACKEN disabled; STRICT disabled; STREN enabled; GCEN disabled;
SMEN disabled; DISSLW enabled; I2CSIDL disabled; ACKDT Sends ACK;
SCLREL Holds; RSEN disabled; A10M 7 Bit; PEN disabled; RCEN disabled;
SEN disabled; I2CEN enabled;
I2C1CONL = 0x8040;
// BCL disabled; P disabled; S disabled; I2COV disabled; IWCOL
disabled;
I2C1STAT = 0x00;

/* I2C1 Master Events */
// clear the master interrupt flag
IFS1bits.MI2C1IF = 0;
// enable the master interrupt
IEC1bits.MI2C1IE = 1;
}

uint8_t I2C1_ErrorCountGet(void)
{
    uint8_t ret;

    ret = i2c1_object.i2cErrors;
    return ret;
}

void __attribute__ ( ( interrupt, no_auto_psv ) ) _MI2C1Interrupt (
void )
{
    static uint8_t *pi2c_buf_ptr;
    static uint16_t i2c_address;
    static uint8_t i2c_bytes_left;
    static uint8_t i2c_10bit_address_restart = 0;

    IFS1bits.MI2C1IF = 0;

    // Check first if there was a collision.
    // If we have a Write Collision, reset and go to idle state */
    if(I2C1_WRITE_COLLISION_STATUS_BIT)
    {
        // clear the Write colision
        I2C1_WRITE_COLLISION_STATUS_BIT = 0;
        i2c1_state = S_MASTER_IDLE;
        *(p_i2c1_current->pTrFlag) = I2C1_MESSAGE_FAIL;

        // reset the buffer pointer
        p_i2c1_current = NULL;

        return;
    }

    /* Handle the correct i2c state */
    switch(i2c1_state)
    {
        case S_MASTER_IDLE: /* In reset state, waiting for data to
send */
```

```

if(i2c1_object.trStatus.s.empty != true)
{
    // grab the item pointed by the head
    p_i2c1_current      = i2c1_object.pTrHead;
    i2c1_trb_count      = i2c1_object.pTrHead->count;
    p_i2c1_trb_current = i2c1_object.pTrHead->ptrb_list;

    i2c1_object.pTrHead++;

    // check if the end of the array is reached
    if(i2c1_object.pTrHead == (i2c1_tr_queue +
I2C1_CONFIG_TR_QUEUE_LENGTH))
    {
        // adjust to restart at the beginning of the array
        i2c1_object.pTrHead = i2c1_tr_queue;
    }

    // since we moved one item to be processed, we know
    // it is not full, so set the full status to false
    i2c1_object.trStatus.s.full = false;

    // check if the queue is empty
    if(i2c1_object.pTrHead == i2c1_object.pTrTail)
    {
        // it is empty so set the empty status to true
        i2c1_object.trStatus.s.empty = true;
    }

    // send the start condition
    I2C1_START_CONDITION_ENABLE_BIT = 1;

    // start the i2c request
    i2c1_state = S_MASTER_SEND_ADDR;
}

break;

case S_MASTER_RESTART:

    /* check for pending i2c Request */

    // ... trigger a REPEATED START
    I2C1_REPEAT_START_CONDITION_ENABLE_BIT = 1;

    // start the i2c request
    i2c1_state = S_MASTER_SEND_ADDR;

    break;

case S_MASTER_SEND_ADDR_10BIT_LSB:

    if(I2C1_ACKNOWLEDGE_STATUS_BIT)
    {
        i2c1_object.i2cErrors++;
        I2C1_Stop(I2C1_MESSAGE_ADDRESS_NO_ACK);
    }
    else
    {
        // Remove bit 0 as R/W is never sent here
        I2C1_TRANSMIT_REG = (i2c_address >> 1) & 0x00FF;
    }
}

```

```

        // determine the next state, check R/W
        if(i2c_address & 0x01)
        {
            // if this is a read we must repeat start
            // the bus to perform a read
            i2c1_state = S_MASTER_10BIT_RESTART;
        }
        else
        {
            // this is a write continue writing data
            i2c1_state = S_MASTER_SEND_DATA;
        }
    }

    break;

case S_MASTER_10BIT_RESTART:

    if(I2C1_ACKNOWLEDGE_STATUS_BIT)
    {
        i2c1_object.i2cErrors++;
        I2C1_Stop(I2C1_MESSAGE_ADDRESS_NO_ACK);
    }
    else
    {
        // ACK Status is good
        // restart the bus
        I2C1_REPEAT_START_CONDITION_ENABLE_BIT = 1;

        // fudge the address so S_MASTER_SEND_ADDR works
        // we only do this on a 10-bit address resend
        i2c_address = 0x00F0 | ((i2c_address >> 8) & 0x0006);

        // set the R/W flag
        i2c_address |= 0x0001;

        // set the address restart flag so we do not change
        i2c_10bit_address_restart = 1;

        // Resend the address as a read
        i2c1_state = S_MASTER_SEND_ADDR;
    }

    break;

case S_MASTER_SEND_ADDR:

    /* Start has been sent, send the address byte */

    /* Note:
       On a 10-bit address resend (done only during a 10-bit
       device read), the original i2c_address was modified in
       S_MASTER_10BIT_RESTART state. So the check if this is
       a 10-bit address will fail and a normal 7-bit address
       is sent with the R/W bit set to read. The flag
       i2c_10bit_address_restart prevents the address to
       be re-written.
    */
    if(i2c_10bit_address_restart != 1)

```

```

    {
        // extract the information for this message
        i2c_address    = p_i2c1_trb_current->address;
        pi2c_buf_ptr   = p_i2c1_trb_current->pbuffer;
        i2c_bytes_left = p_i2c1_trb_current->length;
    }
    else
    {
        // reset the flag so the next access is ok
        i2c_10bit_address_restart = 0;
    }

    // check for 10-bit address
    if(i2c_address > 0x00FF)
    {
        // we have a 10 bit address
        // send bits<9:8>
        // mask bit 0 as this is always a write
        I2C1_TRANSMIT_REG = 0xF0 | ((i2c_address >> 8) &
0x0006);

        i2c1_state = S_MASTER_SEND_ADDR_10BIT_LSB;
    }
    else
    {
        // Transmit the address
        I2C1_TRANSMIT_REG = i2c_address;
        if(i2c_address & 0x01)
        {
            // Next state is to wait for address to be acked
            i2c1_state = S_MASTER_ACK_ADDR;
        }
        else
        {
            // Next state is transmit
            i2c1_state = S_MASTER_SEND_DATA;
        }
    }
    break;

case S_MASTER_SEND_DATA:

    // Make sure the previous byte was acknowledged
    if(I2C1_ACKNOWLEDGE_STATUS_BIT)
    {
        // Transmission was not acknowledged
        i2c1_object.i2cErrors++;

        // Reset the Ack flag
        I2C1_ACKNOWLEDGE_STATUS_BIT = 0;

        // Send a stop flag and go back to idle
        I2C1_Stop(I2C1_DATA_NO_ACK);
    }
    else
    {
        // Did we send them all ?
        if(i2c_bytes_left-- == 0U)
        {
            // yup sent them all!

```

```

        // update the trb pointer
        p_i2c1_trb_current++;

        // are we done with this string of requests?
        if(--i2c1_trb_count == 0)
        {
            I2C1_Stop(I2C1_MESSAGE_COMPLETE);
        }
        else
        {
            // no!, there are more TRB to be sent.
            //I2C1_START_CONDITION_ENABLE_BIT = 1;

            // In some cases, the slave may require
            // a restart instead of a start. So use this
            one

            // instead.
            I2C1_REPEAT_START_CONDITION_ENABLE_BIT = 1;

            // start the i2c request
            i2c1_state = S_MASTER_SEND_ADDR;

        }
    }
    else
    {
        // Grab the next data to transmit
        I2C1_TRANSMIT_REG = *pi2c_buf_ptr++;
    }
}
break;

case S_MASTER_ACK_ADDR:

    /* Make sure the previous byte was acknowledged */
    if(I2C1_ACKNOWLEDGE_STATUS_BIT)
    {
        // Transmission was not acknowledged
        i2c1_object.i2cErrors++;

        // Send a stop flag and go back to idle
        I2C1_Stop(I2C1_MESSAGE_ADDRESS_NO_ACK);

        // Reset the Ack flag
        I2C1_ACKNOWLEDGE_STATUS_BIT = 0;
    }
    else
    {
        I2C1_RECEIVE_ENABLE_BIT = 1;
        i2c1_state = S_MASTER_ACK_RCV_DATA;
    }
    break;

case S_MASTER_RCV_DATA:

    /* Acknowledge is completed. Time for more data */

    // Next thing is to ack the data
    i2c1_state = S_MASTER_ACK_RCV_DATA;

```

```
// Set up to receive a byte of data
I2C1_RECEIVE_ENABLE_BIT = 1;

break;

case S_MASTER_ACK_RCV_DATA:

    // Grab the byte of data received and acknowledge it
    *pi2c_buf_ptr++ = I2C1_RECEIVE_REG;

    // Check if we received them all?
    if(--i2c_bytes_left)
    {

        /* No, there's more to receive */

        // No, bit 7 is clear. Data is ok
        // Set the flag to acknowledge the data
        I2C1_ACKNOWLEDGE_DATA_BIT = 0;

        // Wait for the acknowledge to complete, then get more
        i2c1_state = S_MASTER_RCV_DATA;
    }
    else
    {

        // Yes, it's the last byte. Don't ack it
        // Flag that we will nak the data
        I2C1_ACKNOWLEDGE_DATA_BIT = 1;

        I2C1_FunctionComplete();
    }

    // Initiate the acknowledge
    I2C1_ACKNOWLEDGE_ENABLE_BIT = 1;
    break;

case S_MASTER_RCV_STOP:
case S_MASTER_SEND_STOP:

    // Send the stop flag
    I2C1_Stop(I2C1_MESSAGE_COMPLETE);
    break;

default:

    // This case should not happen, if it does then
    // terminate the transfer
    i2c1_object.i2cErrors++;
    I2C1_Stop(I2C1_LOST_STATE);
    break;
}
}

static void I2C1_FunctionComplete(void)
{

    // update the trb pointer
    p_i2c1_trb_current++;
}
```

```
        // are we done with this string of requests?
        if(--i2c1_trb_count == 0)
        {
            i2c1_state = S_MASTER_SEND_STOP;
        }
        else
        {
            i2c1_state = S_MASTER_RESTART;
        }
    }

static void I2C1_Stop(I2C1_MESSAGE_STATUS completion_code)
{
    // then send a stop
    I2C1_STOP_CONDITION_ENABLE_BIT = 1;

    // make sure the flag pointer is not NULL
    if (p_i2c1_current->pTrFlag != NULL)
    {
        // update the flag with the completion code
        *(p_i2c1_current->pTrFlag) = completion_code;
    }

    // Done, back to idle
    i2c1_state = S_MASTER_IDLE;
}

void I2C1_MasterWrite(
                                uint8_t *pdata,
                                uint8_t length,
                                uint16_t address,
                                I2C1_MESSAGE_STATUS *pstatus)
{
    static I2C1_TRANSACTION_REQUEST_BLOCK    trBlock;

    // check if there is space in the queue
    if (i2c1_object.trStatus.s.full != true)
    {
        I2C1_MasterWriteTRBBuild(&trBlock, pdata, length, address);
        I2C1_MasterTRBInsert(1, &trBlock, pstatus);
    }
    else
    {
        *pstatus = I2C1_MESSAGE_FAIL;
    }
}

void I2C1_MasterRead(
                                uint8_t *pdata,
                                uint8_t length,
                                uint16_t address,
                                I2C1_MESSAGE_STATUS *pstatus)
{
    static I2C1_TRANSACTION_REQUEST_BLOCK    trBlock;

    // check if there is space in the queue
    if (i2c1_object.trStatus.s.full != true)
```

```

        {
            I2C1_MasterReadTRBBuild(&trBlock, pdata, length, address);
            I2C1_MasterTRBInsert(1, &trBlock, pstatus);
        }
    else
    {
        *pstatus = I2C1_MESSAGE_FAIL;
    }
}

void I2C1_MasterTRBInsert(
                                uint8_t count,
                                I2C1_TRANSACTION_REQUEST_BLOCK
    *ptrb_list,
                                I2C1_MESSAGE_STATUS *pflag)
{
    // check if there is space in the queue
    if (i2c1_object.trStatus.s.full != true)
    {
        *pflag = I2C1_MESSAGE_PENDING;

        i2c1_object.pTrTail->ptrb_list = ptrb_list;
        i2c1_object.pTrTail->count     = count;
        i2c1_object.pTrTail->pTrFlag   = pflag;
        i2c1_object.pTrTail++;

        // check if the end of the array is reached
        if (i2c1_object.pTrTail == (i2c1_tr_queue +
I2C1_CONFIG_TR_QUEUE_LENGTH))
        {
            // adjust to restart at the beginning of the array
            i2c1_object.pTrTail = i2c1_tr_queue;
        }

        // since we added one item to be processed, we know
        // it is not empty, so set the empty status to false
        i2c1_object.trStatus.s.empty = false;

        // check if full
        if (i2c1_object.pTrHead == i2c1_object.pTrTail)
        {
            // it is full, set the full status to true
            i2c1_object.trStatus.s.full = true;
        }

        // for interrupt based
        if(i2c1_state == S_MASTER_IDLE)
        {
            // force the task to run since we know that the queue has
            // something that needs to be sent
            IFS1bits.MI2C1IF = 1;
        }
    }
    else
    {
        *pflag = I2C1_MESSAGE_FAIL;
    }
}

```

```
}

void I2C1_MasterReadTRBBuild(
                                I2C1_TRANSACTION_REQUEST_BLOCK *ptrb,
                                uint8_t *pdata,
                                uint8_t length,
                                uint16_t address)
{
    ptrb->address = address << 1;
    // make this a read
    ptrb->address |= 0x01;
    ptrb->length = length;
    ptrb->pbuffer = pdata;
}

void I2C1_MasterWriteTRBBuild(
                                I2C1_TRANSACTION_REQUEST_BLOCK *ptrb,
                                uint8_t *pdata,
                                uint8_t length,
                                uint16_t address)
{
    ptrb->address = address << 1;
    ptrb->length = length;
    ptrb->pbuffer = pdata;
}

bool I2C1_MasterQueueIsEmpty(void)
{
    return((bool)i2c1_object.trStatus.s.empty);
}

bool I2C1_MasterQueueIsFull(void)
{
    return((bool)i2c1_object.trStatus.s.full);
}

/**
    End of File
*/
```

### 3.2.5. pin\_manager.c

```
#include <xc.h>
#include <stdio.h>
#include "pin_manager.h"
#include "pwm.h"
#include "Variables.h"

/**
 * Section: File specific functions
 */
void (*CN_InterruptHandler)(void) = NULL;

/**
 * Section: Driver Interface Function Definitions
 */
void PIN_MANAGER_Initialize (void)
{
    /*****
     * Setting the Output Latch SFR(s)
     *****/
    LATA = 0x0000;
    LATB = 0x0060;

    /*****
     * Setting the GPIO Direction SFR(s)
     *****/
    TRISA = 0x001F;
    TRISB = 0x037F;

    /*****
     * Setting the Weak Pull Up and Weak Pull Down SFR(s)
     *****/
    CNPDA = 0x0000;
    CNPDB = 0x0000;
    CNPUA = 0x0000;
    CNPUB = 0x0060;

    /*****
     * Setting the Open Drain SFR(s)
     *****/
    ODCA = 0x0000;
    ODCB = 0x0000;
```

```

/*****
* Setting the Analog/Digital Configuration SFR(s)
*****/
ANSELA = 0x0003;
ANSELB = 0x0203;

/*****
* Interrupt On Change: any
*****/
CNENBbits.CNIEB8 = 1;    //Pin : RB8

/*****
* Interrupt On Change: Interrupt Enable
*****/
IFS1bits.CNIF = 0; //Clear CNI interrupt flag
IEC1bits.CNIE = 1; //Enable CNI interrupt
}

void CN_SetInterruptHandler(void (* InterruptHandler) (void))
{
    IEC1bits.CNIE = 0; //Disable CNI interrupt
    CN_InterruptHandler = InterruptHandler;
    IEC1bits.CNIE = 1; //Enable CNI interrupt
}

/* Interrupt service routine for the CNI interrupt. */
void __attribute__(( interrupt, no_auto_psv )) _CNInterrupt ( void )
{
    if(IFS1bits.CNIF == 1)
    {
        // Clear the flag
        IFS1bits.CNIF = 0;
        if(CN_InterruptHandler)
        {
            CN_InterruptHandler();
        }
    }

    if (PORTBbits.RB8== 1)    // Flanco ascendente
    {
        Temp = 1;
        PWM_ModuleDisable(); // Se apaga el módulo PWM
        OUT = 0;             // Se desactiva la salida
    }
    else                    // Flanco descendente
    {
        Temp = 0;
    }
}

```

```
        PWM_ModuleEnable();    // Se reactiva el módulo PWM
    }
}
```

### 3.2.6. pwm.c

```
#include "pwm.h"

/**
 * Section: Driver Interface
 */

void PWM_Initialize (void)
{
    // PCLKDIV 1;
    PTCON2 = 0x00;
    // PTPER 1200;
    PTPER = 0x4B0;
    // SEVTCMP 0;
    SEVTCMP = 0x00;
    // MDC 600;
    MDC = 0x258;
    // CHOPCLK 0; CHPCLKEN disabled;
    CHOP = 0x00;
    // PWMKEY 0;
    PWMKEY = 0x00;
    // MDCS Master; FLTIEN disabled; CAM Edge Aligned; DTC Positive
    dead time for all Output modes; TRGIEN disabled; XPRES disabled; ITB
    Master; IUE disabled; CLIEN disabled; DTCP disabled;
    PWMCON1 = 0x100;
    // MDCS Master; FLTIEN disabled; CAM Edge Aligned; DTC Positive
    dead time for all Output modes; TRGIEN disabled; XPRES disabled; ITB
    Master; IUE disabled; CLIEN disabled; DTCP disabled;
    PWMCON2 = 0x100;
    // MDCS Primary; FLTIEN disabled; CAM Edge Aligned; DTC Positive
    dead time for all Output modes; TRGIEN disabled; XPRES disabled; ITB
    Master; IUE disabled; CLIEN disabled; DTCP disabled;
    PWMCON3 = 0x00;
    //FLTDAT PWM1L Low, PWM1H Low; SWAP disabled; OVRENH disabled;
    PENL enabled; PMOD Complementary Output Mode; OVRENL disabled; OSYNC
    disabled; POLL disabled; PENH enabled; CLDAT PWM1L Low, PWM1H Low;
    OVRDAT PWM1L Low, PWM1H Low; POLH disabled;
    __builtin_write_PWMSFR(&IOCON1, 0xC000, &PWMKEY);
    //FLTDAT PWM2L Low, PWM2H Low; SWAP disabled; OVRENH disabled;
    PENL enabled; PMOD Complementary Output Mode; OVRENL disabled; OSYNC
    disabled; POLL disabled; PENH enabled; CLDAT PWM2L Low, PWM2H Low;
    OVRDAT PWM2L Low, PWM2H Low; POLH disabled;
    __builtin_write_PWMSFR(&IOCON2, 0xC000, &PWMKEY);
    //FLTDAT PWM3L Low, PWM3H Low; SWAP disabled; OVRENH disabled;
    PENL enabled; PMOD Complementary Output Mode; OVRENL disabled; OSYNC
    disabled; POLL disabled; PENH enabled; CLDAT PWM3L Low, PWM3H Low;
    OVRDAT PWM3L Low, PWM3H Low; POLH disabled;
    __builtin_write_PWMSFR(&IOCON3, 0xC000, &PWMKEY);
    //FLTPOL disabled; CLPOL disabled; CLSRC FLT1; CLMOD disabled;
    FLTMOD PWM1H, PWM1L pins to FLTDAT values- Latched; IFLTMOD disabled;
    FLTSRC FLT32;
    __builtin_write_PWMSFR(&FCLCON1, 0xF8, &PWMKEY);
    //FLTPOL disabled; CLPOL disabled; CLSRC FLT1; CLMOD disabled;
    FLTMOD PWM2H, PWM2L pins to FLTDAT values- Latched; IFLTMOD disabled;
    FLTSRC FLT32;
    __builtin_write_PWMSFR(&FCLCON2, 0xF8, &PWMKEY);
}
```

```
//FLTPOL disabled; CLPOL disabled; CLSRC FLT1; CLMOD disabled;
FLTMOD PWM3H, PWM3L pins to FLTDAT values- Latched; IFLTMOD disabled;
FLTSRC FLT32;
    builtin_write_PWMSFR(&FCLCON3, 0xF8, &PWMKEY);
// PDC1 0;
PDC1 = 0x00;
// PDC2 0;
PDC2 = 0x00;
// PDC3 0;
PDC3 = 0x00;
// PHASE1 0;
PHASE1 = 0x00;
// PHASE2 0;
PHASE2 = 0x00;
// PHASE3 0;
PHASE3 = 0x00;
// DTR1 3;
DTR1 = 0x03;
// DTR2 3;
DTR2 = 0x03;
// DTR3 0;
DTR3 = 0x00;
// ALTDTR1 3;
ALTDTR1 = 0x03;
// ALTDTR2 3;
ALTDTR2 = 0x03;
// ALTDTR3 0;
ALTDTR3 = 0x00;
// TRGCMP 0;
TRIG1 = 0x00;
// TRGCMP 0;
TRIG2 = 0x00;
// TRGCMP 0;
TRIG3 = 0x00;
// TRGDIV 1; TRGSTRT 0;
TRGCON1 = 0x00;
// TRGDIV 1; TRGSTRT 0;
TRGCON2 = 0x00;
// TRGDIV 1; TRGSTRT 0;
TRGCON3 = 0x00;
// PWMCAP 0;
PWMCAP1 = 0x00;
// PWMCAP 0;
PWMCAP2 = 0x00;
// PWMCAP 0;
PWMCAP3 = 0x00;
// BPLL disabled; BPHH disabled; BPLH disabled; BCH disabled;
FLTLEBEN disabled; PLR disabled; CLLEBEN disabled; BCL disabled; PLF
disabled; PHR disabled; BPHL disabled; PHF disabled;
LEBCON1 = 0x00;
// BPLL disabled; BPHH disabled; BPLH disabled; BCH disabled;
FLTLEBEN disabled; PLR disabled; CLLEBEN disabled; BCL disabled; PLF
disabled; PHR disabled; BPHL disabled; PHF disabled;
LEBCON2 = 0x00;
// BPLL disabled; BPHH disabled; BPLH disabled; BCH disabled;
FLTLEBEN disabled; PLR disabled; CLLEBEN disabled; BCL disabled; PLF
disabled; PHR disabled; BPHL disabled; PHF disabled;
LEBCON3 = 0x00;
// LEB 0;
LEBDLY1 = 0x00;
// LEB 0;
```

```
LEBDLY2 = 0x00;
// LEB 0;
LEBDLY3 = 0x00;
// CHOPLEN disabled; CHOPHEN disabled; BLANKSEL No state blanking;
CHOPSEL No state blanking;
AUXCON1 = 0x00;
// CHOPLEN disabled; CHOPHEN disabled; BLANKSEL No state blanking;
CHOPSEL No state blanking;
AUXCON2 = 0x00;
// CHOPLEN disabled; CHOPHEN disabled; BLANKSEL No state blanking;
CHOPSEL No state blanking;
AUXCON3 = 0x00;

// SYNCOEN disabled; SEIEN disabled; SESTAT disabled; SEVTPS 1;
SYNCSRC SYNCI1; SYNCEN disabled; PTSIDL disabled; PTEN enabled; EIPU
disabled; SYNCPOL disabled;
PTCON = 0x8000;
}

void __attribute__((weak)) PWM_SpecialEvent_CallBack(void)
{
    // Add your custom callback code here
}

void PWM_SpecialEvent_Tasks ( void )
{
    if(IFS3bits.PSEMIF)
    {
        // PWM SpecialEvent callback function
        PWM_SpecialEvent_CallBack();

        // clear the PWM SpecialEvent interrupt flag
        IFS3bits.PSEMIF = 0;
    }
}

void __attribute__((weak)) PWM_Generator1_CallBack(void)
{
    // Add your custom callback code here
}

void PWM_Generator1_Tasks ( void )
{
    if(IFS3bits.PSEMIF)
    {
        // PWM Generator1 callback function
        PWM_Generator1_CallBack();

        // clear the PWM Generator1 interrupt flag
        IFS5bits.PWM1IF = 0;
    }
}

void __attribute__((weak)) PWM_Generator2_CallBack(void)
{
    // Add your custom callback code here
}

void PWM_Generator2_Tasks ( void )
{
    if(IFS3bits.PSEMIF)
```

```
    {
        // PWM Generator2 callback function
        PWM_Generator2_CallBack();

        // clear the PWM Generator2 interrupt flag
        IFS5bits.PWM2IF = 0;
    }
}
void __attribute__((weak)) PWM_Generator3_CallBack(void)
{
    // Add your custom callback code here
}

void PWM_Generator3_Tasks ( void )
{
    if(IFS3bits.PSEMIIF)
    {
        // PWM Generator3 callback function
        PWM_Generator3_CallBack();

        // clear the PWM Generator3 interrupt flag
        IFS6bits.PWM3IF = 0;
    }
}

/**
End of File
*/
```

### 3.2.7. reset.c

```
#include <stdbool.h>
#include <stdint.h>
#include "xc.h"
#include "reset.h"

/**
 * Section: Local Variables
 */

/**
 * Section: Function prototypes
 */
static bool RESET_CauseFromSoftware(uint16_t resetCause);
static bool RESET_CauseFromWatchdogTimer(uint16_t resetCause);
static bool RESET_CauseFromConfigurationMismatch(uint16_t resetCause);
static bool RESET_CauseFromIllegalOpcode(uint16_t resetCause);
static bool RESET_CauseFromExternal(uint16_t resetCause);
static bool RESET_CauseFromTrap(uint16_t resetCause);
static void RESET_CauseClear(RESET_MASKS resetFlagMask);

uint16_t RESET_GetCause(void)
{
    return RCON;
}

void __attribute__((weak)) RESET_CauseHandler(void)
{
    uint16_t resetCause = RESET_GetCause();
    if(RESET_CauseFromTrap(resetCause))
    {
        RESET_CauseClear(RESET_MASK_TRAPR);
        //Do something
    }
    if(RESET_CauseFromIllegalOpcode(resetCause))
    {
        RESET_CauseClear(RESET_MASK_IOPUWR);
        //Do something
    }
    if(RESET_CauseFromConfigurationMismatch(resetCause))
    {
        RESET_CauseClear(RESET_MASK_CM);
        //Do something
    }
    if(RESET_CauseFromExternal(resetCause))
    {
        RESET_CauseClear(RESET_MASK_EXTR);
        //Do something
    }
    if(RESET_CauseFromSoftware(resetCause))
    {
        RESET_CauseClear(RESET_MASK_SWR);
        //Do something
    }
    if(RESET_CauseFromWatchdogTimer(resetCause))
    {
        RESET_CauseClear(RESET_MASK_WDTO);
        //Do something
    }
}
```

```
static bool RESET_CauseFromTrap(uint16_t resetCause)
{
    bool resetStatus = false;
    if(resetCause & RESET_MASK_TRAPR)
    {
        resetStatus = true;
    }
    return resetStatus;
}

static bool RESET_CauseFromIllegalOpcode(uint16_t resetCause)
{
    bool resetStatus = false;
    if(resetCause & RESET_MASK_IOPUWR)
    {
        resetStatus = true;
    }
    return resetStatus;
}

static bool RESET_CauseFromConfigurationMismatch(uint16_t resetCause)
{
    bool resetStatus = false;
    if(resetCause & RESET_MASK_CM)
    {
        resetStatus = true;
    }
    return resetStatus;
}

static bool RESET_CauseFromExternal(uint16_t resetCause)
{
    bool resetStatus = false;
    if(resetCause & RESET_MASK_EXTR)
    {
        resetStatus = true;
    }
    return resetStatus;
}

static bool RESET_CauseFromSoftware(uint16_t resetCause)
{
    bool resetStatus = false;
    if(resetCause & RESET_MASK_SWR)
    {
        resetStatus = true;
    }
    return resetStatus;
}

static bool RESET_CauseFromWatchdogTimer(uint16_t resetCause)
{
    bool resetStatus = false;
    if(resetCause & RESET_MASK_WDTO)
    {
        resetStatus = true;
    }
    return resetStatus;
}
```

```
static void RESET_CauseClear(RESET_MASKS resetFlagMask)
{
    RCON = RCON & (~resetFlagMask);
}

void RESET_CauseClearAll()
{
    RCON = 0x00;
}
/**
End of File
*/
```

### 3.2.8. System.c

```
#include "pin_manager.h"
#include "clock.h"
#include "system.h"
#include "system_types.h"
#include "i2c1.h"
#include "pwm.h"
#include "tmr1.h"
#include "adc1.h"
#include "interrupt_manager.h"
#include "traps.h"

void SYSTEM_Initialize(void)
{
    PIN_MANAGER_Initialize();
    INTERRUPT_Initialize();
    CLOCK_Initialize();
    I2C1_Initialize();
    ADC1_Initialize();
    PWM_Initialize();
    TMR1_Initialize();
    INTERRUPT_GlobalEnable();
    SYSTEM_CORCONModeOperatingSet(CORCON_MODE_PORVALUES);
}

/**
End of File
*/
```

### 3.2.9. tmr1.c

```
#define SCALE 1L<<15 /* 2^15 */
#define RANGE 1L<<16 /* 2^16 */

#include <stdio.h>
#include "tmr1.h"
#include "i2c1.h"
#include "string.h"
#include <stdlib.h>
#include "Variables.h"
#include "math.h"
#include <dsp.h>

static I2C1_MESSAGE_STATUS status;
static I2C1_MESSAGE_STATUS status1;
uint8_t longitudW = 16;
uint16_t address = 0x01;
uint8_t longitudR = 16;
uint16_t address1 = 0x01;
char STemp[1];
char SCCC[1];
int status2;

/**
 * Section: File specific functions
 */
void (*TMR1_InterruptHandler)(void) = NULL;
void TMR1_CallBack(void);

/**
 * Section: Data Type Definitions
 */

/** TMR Driver Hardware Instance Object

 @Summary
 Defines the object required for the maintenance of the hardware
 instance.

 @Description
 This defines the object required for the maintenance of the
 hardware
 instance. This object exists once per hardware instance of the
 peripheral.

 Remarks:
 None.
 */

typedef struct _TMR_OBJ_STRUCT
{
    /* Timer Elapsed */
    volatile bool timerElapsed;
    /*Software Counter value*/
    volatile uint8_t count;
} TMR_OBJ;
```

```
static TMR_OBJ tmr1_obj;

/**
 * Section: Driver Interface
 */

void TMR1_Initialize (void)
{
    //TMR1 0;
    TMR1 = 0x00;
    //Period = 0.09999978667 s; Frequency = 60000000 Hz; PR1 23436;
    PR1 = 0x5B8C;
    //TCKPS 1:256; TON enabled; TSIDL disabled; TCS FOSC/2; TSYNC
    disabled; TGATE disabled;
    T1CON = 0x8030;

    if(TMR1_InterruptHandler == NULL)
    {
        TMR1_SetInterruptHandler(&TMR1_CallBack);
    }

    IFS0bits.T1IF = false;
    IEC0bits.T1IE = true;

    tmr1_obj.timerElapsed = false;
}

void __attribute__ ( ( interrupt, no_auto_psv ) ) _T1Interrupt ( )
{
    /* Check if the Timer Interrupt/Status is set */

    /***User Area Begin

    // ticker function call;
    // ticker is 1 -> Callback function gets called everytime this ISR
    executes
    if(TMR1_InterruptHandler)
    {
        TMR1_InterruptHandler();
    }

    /***User Area End

    tmr1_obj.count++;
    tmr1_obj.timerElapsed = true;
    IFS0bits.T1IF = false;
}

void TMR1_Period16BitSet( uint16_t value )
{
    /* Update the counter values */
    PR1 = value;
    /* Reset the status information */
    tmr1_obj.timerElapsed = false;
}

uint16_t TMR1_Period16BitGet( void )
{
    return( PR1 );
}
```

```
}

void TMR1_Counter16BitSet ( uint16_t value )
{
    /* Update the counter values */
    TMR1 = value;
    /* Reset the status information */
    tmr1_obj.timerElapsed = false;
}

uint16_t TMR1_Counter16BitGet( void )
{
    return( TMR1 );
}

void __attribute__ ((weak)) TMR1_CallBack(void)
{
    // Add your custom callback code here
    if (LeerEscribir==0)
    {
        // Escalado de medidas float al rango completo de medida
        VoltajeSecLectura = (VSecQ12*65.08) /4096;
        CorrienteSecLectura = (CorrienteSec*22.676) /4096;
        CorrientePrimLectura = (CorrientePrim*4.65) /4096;

        // Conversión a String
        ftoa(VoltajeSecLectura, VoltajeSecLecturaString, 2);
        ftoa(CorrienteSecLectura, CorrienteSecLecturaString, 2);

        if (VoltajeSecLectura < 1.00){
            V1[0] = '0';
            V1[1] = '0';
            V1[2] = VoltajeSecLecturaString[0];
            V1[3] = VoltajeSecLecturaString[1];
            V1[4] = VoltajeSecLecturaString[2];
        }
        else if (VoltajeSecLectura < 10.00)
        {
            V1[0] = '0';
            V1[1] = VoltajeSecLecturaString[0];
            V1[2] = VoltajeSecLecturaString[1];
            V1[3] = VoltajeSecLecturaString[2];
            V1[4] = VoltajeSecLecturaString[3];
        }
        else
        {
            V1[0] = VoltajeSecLecturaString[0];
            V1[1] = VoltajeSecLecturaString[1];
            V1[2] = VoltajeSecLecturaString[2];
            V1[3] = VoltajeSecLecturaString[3];
            V1[4] = VoltajeSecLecturaString[4];
        }
    }

    if (CorrienteSecLectura < 1.00)
    {
        V1[6] = '0';
        V1[7] = '0';
        V1[8] = CorrienteSecLecturaString[0];
        V1[9] = CorrienteSecLecturaString[1];
        V1[10] = CorrienteSecLecturaString[2];
    }
}
```

```

    }
    else if (CorrienteSecLectura < 10.00){
        V1[6] = '0';
        V1[7] = CorrienteSecLecturaString[0];
        V1[8] = CorrienteSecLecturaString[1];
        V1[9] = CorrienteSecLecturaString[2];
        V1[10] = CorrienteSecLecturaString[3];
    }
    else
    {
        V1[6] = CorrienteSecLecturaString[0];
        V1[7] = CorrienteSecLecturaString[1];
        V1[8] = CorrienteSecLecturaString[2];
        V1[9] = CorrienteSecLecturaString[3];
        V1[10] = CorrienteSecLecturaString[4];
    }

    V1[5] = ',';
    V1[11] = ',';
    intToStr(CCCV, SCCC, 1);
    V1[12] = SCCC[0];
    V1[13] = ',';
    intToStr(Temp, STemp, 1);
    V1[14]=STemp[0];
    V1[15] = '\n';

    I2C1_MasterWrite(&V1, longitudW, address, &status);
    LeerEscribir=1;
}
else
{
    I2C1_MasterRead(&Recibo, longitudR ,address1, &status1);

    memcpy(VString, &Recibo[0], 5);
    memcpy(CorrienteString, &Recibo[6], 5);
    memcpy(&OUT, &Recibo[12], 1);
    // Referencias para controlador
    VoltajeConsignaFloat = atof(VString);
    CorrienteConsignaFloat = atof(CorrienteString);
    LeerEscribir=0;

    CorrientePrimRef =
    ((CorrienteConsignaFloat/4.5)*32768)/3.555555;           //Corriente
    Secundario partido de la relacion de transformacion

    // Escalamos las consignas de Float a formato Q15. Siendo el
    valor maximo 60V = 32768
    VoltajeConsignaQ12= (VoltajeConsignaFloat*4096)/65.08;
    CorrienteConsignaQ12 = (CorrienteConsignaFloat*4096)/22.67;

    // Calculamos la rampa a corregir en el controlador. D =
    Vout/Vin. Ri = 4V/3.72A = 0.86. Ts = 1/60000 Hz
    Vpp = -((0.18-
    (VoltajeConsignaFloat/400))*0.86*0.00001666*400)/0.000033;

    // Convertimos a un valor digital segun la escala del DAC
    Ramp = Vpp*32768/5;

    // Incremento a compensar
    DeltaRamp = - (Ramp*60000)/60000000;

```

```
    }

}

void TMR1_SetInterruptHandler(void (* InterruptHandler) (void))
{
    IEC0bits.T1IE = false;
    TMR1_InterruptHandler = InterruptHandler;
    IEC0bits.T1IE = true;
}

void TMR1_Start( void )
{
    /* Reset the status information */
    tmr1_obj.timerElapsed = false;

    /*Enable the interrupt*/
    IEC0bits.T1IE = true;

    /* Start the Timer */
    T1CONbits.TON = 1;
}

void TMR1_Stop( void )
{
    /* Stop the Timer */
    T1CONbits.TON = false;

    /*Disable the interrupt*/
    IEC0bits.T1IE = false;
}

bool TMR1_GetElapsedThenClear(void)
{
    bool status;

    status = tmr1_obj.timerElapsed;

    if(status == true)
    {
        tmr1_obj.timerElapsed = false;
    }
    return status;
}

int TMR1_SoftwareCounterGet(void)
{
    return tmr1_obj.count;
}

void TMR1_SoftwareCounterClear(void)
{
    tmr1_obj.count = 0;
}

/**
    End of File
*/

// Converts a floating point number to string.
```

```
void ftoa(float n, char *res, int afterpoint)
{
    // Extract integer part
    int ipart = (int)n;

    // Extract floating part
    float fpart = n - (float)ipart;

    // convert integer part to string
    int i = intToStr(ipart, res, 0);

    // check for display option after point
    if (afterpoint != 0)
    {
        res[i] = '.'; // add dot

        // Get the value of fraction part upto given no.
        // of points after dot. The third parameter is needed
        // to handle cases like 233.007
        fpart = fpart * pow(10, afterpoint);

        intToStr((int)fpart, res + i + 1, afterpoint);
    }
}
// reverses a string 'str' of length 'len'
void reverse(char *str, int len)
{
    int i=0, j=len-1, temp;
    while (i<j)
    {
        temp = str[i];
        str[i] = str[j];
        str[j] = temp;
        i++; j--;
    }
}

// Converts a given integer x to string str[]. d is the number
// of digits required in output. If d is more than the number
// of digits in x, then 0s are added at the beginning.
int intToStr(int x, char str[], int d)
{
    int i = 0;
    while (x)
    {
        str[i++] = (x%10) + '0';
        x = x/10;
    }

    // If number of digits required is more, then
    // add 0s at the beginning
    while (i < d)
        str[i++] = '0';

    reverse(str, i);
    str[i] = '\0';
    return i;
}
```

### 3.2.10. traps.c

```
#include <xc.h>
#include "traps.h"

#define ERROR_HANDLER __attribute__((interrupt, no_auto_psv, keep,
section("error_handler")))
#define ERROR_HANDLER_NORETURN ERROR_HANDLER __attribute__((noreturn))
#define FAILSAFE_STACK_GUARDSIZE 8

/**
 * a private place to store the error code if we run into a severe
error
 */
static uint16_t TRAPS_error_code = -1;

/**
 * Halts
 *
 * @param code error code
 */
void __attribute__((naked, noreturn, weak))
TRAPS_halt_on_error(uint16_t code)
{
    TRAPS_error_code = code;
#ifdef __DEBUG
    __builtin_software_breakpoint();
    /* If we are in debug mode, cause a software breakpoint in the
debugger */
#endif
    while(1);
}

/**
 * Sets the stack pointer to a backup area of memory, in case we run
into
 * a stack error (in which case we can't really trust the stack
pointer)
 */
inline static void use_failsafe_stack(void)
{
    static uint8_t failsafe_stack[32];
    asm volatile (
        "    mov    %[pstack], W15\n"
        :
        : [pstack]"r"(failsafe_stack)
    );
    /* Controls where the stack pointer limit is, relative to the end of
the
 * failsafe stack
 */
    SPLIM = (uint16_t)((uint8_t *)failsafe_stack) +
sizeof(failsafe_stack)
- FAILSAFE_STACK_GUARDSIZE);
}

/** Oscillator Fail Trap vector**/
void ERROR_HANDLER_NORETURN _OscillatorFail(void)
```

```
{
    INTCON1bits.OSCFAIL = 0; //Clear the trap flag
    TRAPS_halt_on_error(TRAPS_OSC_FAIL);
}
/** Stack Error Trap Vector**/
void ERROR_HANDLER_NORETURN _StackError(void)
{
    /* We use a failsafe stack: the presence of a stack-pointer error
    * means that we cannot trust the stack to operate correctly
    unless
    * we set the stack pointer to a safe place.
    */
    use_failsafe_stack();
    INTCON1bits.STKERR = 0; //Clear the trap flag
    TRAPS_halt_on_error(TRAPS_STACK_ERR);
}
/** Address error Trap vector**/
void ERROR_HANDLER_NORETURN _AddressError(void)
{
    INTCON1bits.ADDRERR = 0; //Clear the trap flag
    TRAPS_halt_on_error(TRAPS_ADDRESS_ERR);
}
/** Math Error Trap vector**/
void ERROR_HANDLER_NORETURN _MathError(void)
{
    INTCON1bits.MATHERR = 0; //Clear the trap flag
    TRAPS_halt_on_error(TRAPS_MATH_ERR);
}
/** DMAC Error Trap vector**/
void ERROR_HANDLER_NORETURN _DMACError(void)
{
    INTCON1bits.DMACERR = 0; //Clear the trap flag
    TRAPS_halt_on_error(TRAPS_DMAC_ERR);
}
/** Generic Hard Trap vector**/
void ERROR_HANDLER_NORETURN _HardTrapError(void)
{
    INTCON4bits.SGHT = 0; //Clear the trap flag
    TRAPS_halt_on_error(TRAPS_HARD_ERR);
}
/** Generic Soft Trap vector**/
void ERROR_HANDLER_NORETURN _SoftTrapError(void)
{
    INTCON3bits.DOOVR = 0; //Clear the trap flag
    TRAPS_halt_on_error(TRAPS_DOOVR_ERR);
}
```

### 3.2.11. Variables.h

```
/*
 * File:   Variables.h
 * Author: Alberto
 *
 * Created on 22 de julio de 2019, 15:51
 */

#ifndef VARIABLES_H
#define VARIABLES_H

#ifdef __cplusplus
extern "C" {
#endif

#ifdef __cplusplus
}
#endif

#endif /* VARIABLES_H */

extern int Temp;

extern unsigned char V1[16];
//extern unsigned char V2[14];
extern int LeerEscribir;
extern int i;
extern int j;
extern char Recibo[14];
extern char OUT;
extern char CorrienteString[5];
extern char VString[5];
extern int CCCV;

extern float VoltajeConsignaFloat;
extern float CorrienteConsignaFloat;
extern int VoltajeConsignaQ12;
extern int CorrienteConsignaQ12;

extern unsigned int VoltajeSec;
extern int CorrientePrim;
extern int CorrienteSec;

extern float VoltajeSecLectura;
extern float CorrienteSecLectura;
extern float CorrientePrimLectura;

extern char VoltajeSecLecturaString[5];
extern char CorrienteSecLecturaString[5];

extern int VSecQ12;
extern int ISecQ12;
extern int IPrimQ15;

extern int VSecError;
```

```
extern float VSecErrorPrevio;
extern float IPrimError;

// Rampa
extern float Vpp;
extern int Ramp;
extern int Steps;
extern int DeltaRamp;

extern int VoltajeSecMax;
extern int VoltajeSecMin;
extern int CorrienteSecMin;
extern int CorrienteSecMax;
extern int ut;
extern int Fase;
extern int FaseLimite;

extern int IPrimRefQ15;
extern int CorrientePrimRef;
extern int CorrientePrimError;
extern float VL;
extern float SalidaCorriente;
extern float PIDOutput;
extern int DeltaUT;
extern int FaseAnterior;
extern int FaseMax;
extern int FaseMin;
extern int CalculoDocePorCiento;
extern int SalidaControlador;
```

### 3.2.12. Variables.c

```
#include "Variables.h"
#include "libpic30.h"
#include "dsp.h"
#include "xc.h"
#include "stdlib.h"
#include "stdio.h"
#include "math.h"

// Variables ADC
unsigned int VoltajeSec = 0;
int CorrientePrim = 0;
int CorrienteSec = 0;

// Variables I2C
float VoltajeSecLectura;
float CorrienteSecLectura;
float CorrientePrimLectura;
char VoltajeSecLecturaString[5];
char CorrienteSecLecturaString[5];
char Recibo[14];
char OUT;
char CorrienteString[5];
char VString[5];
int CCCV = 0;
int Temp = 0;
unsigned char V1[16];
int LeerEscribir=0;
int i = 0;
int j = 0;

// Consignas para controlador
float VoltajeConsignaFloat;
float CorrienteConsignaFloat;
int VoltajeConsignaQ12;
int CorrienteConsignaQ12;

// Variables de operación controlador
int VSecQ12 = 0;
int ISecQ12 = 0;
int IPrimQ15 = 0;

int VSecError = 0;
float VSecErrorPrevio = 0;
float IPrimError = 0;

int VoltajeSecMax;
int VoltajeSecMin;
int CorrienteSecMin;
int CorrienteSecMax;
int ut = 0;
int Fase = 0;
int FaseLimite = 0;

// Rampa
float Vpp;
int Ramp;
int Steps;
```

```
int DeltaRamp;  
  
int IPrimRefQ15 = 0;  
int CorrientePrimRef = 0;  
int CorrientePrimError = 0;  
float VL = 0;  
float SalidaCorriente = 0;  
float PIDOutput;  
int DeltaUT = 0;  
int FaseAnterior = 0;  
int FaseMax = 576;  
int FaseMin = 0;  
int CalculoDocePorCiento = 0;  
int SalidaControlador;
```

### 3.2.13. main.c

```
/**
Generated main.c file from MPLAB Code Configurator

@Company
  Microchip Technology Inc.

@File Name
  main.c

@Summary
  This is the generated main.c using PIC24 / dsPIC33 / PIC32MM MCUs.

@Description
  This source file provides main entry point for system
  initialization and application code development.
  Generation Information :
    Product Revision : PIC24 / dsPIC33 / PIC32MM MCUs - 1.95-b-
  SNAPSHOT
    Device           : dsPIC33EV256GM002
  The generated drivers are tested against the following:
    Compiler         : XC16 v1.36
    MPLAB            : MPLAB X v5.10
*/

/*
  (c) 2016 Microchip Technology Inc. and its subsidiaries. You may
  use this
  software and any derivatives exclusively with Microchip products.

  THIS SOFTWARE IS SUPPLIED BY MICROCHIP "AS IS". NO WARRANTIES,
  WHETHER
  EXPRESS, IMPLIED OR STATUTORY, APPLY TO THIS SOFTWARE, INCLUDING
  ANY IMPLIED
  WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A
  PARTICULAR PURPOSE, OR ITS INTERACTION WITH MICROCHIP PRODUCTS,
  COMBINATION
  WITH ANY OTHER PRODUCTS, OR USE IN ANY APPLICATION.

  IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL,
  PUNITIVE,
  INCIDENTAL OR CONSEQUENTIAL LOSS, DAMAGE, COST OR EXPENSE OF ANY
  KIND
  WHATSOEVER RELATED TO THE SOFTWARE, HOWEVER CAUSED, EVEN IF
  MICROCHIP HAS
  BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO
  THE
  FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL
  CLAIMS IN
  ANY WAY RELATED TO THIS SOFTWARE WILL NOT EXCEED THE AMOUNT OF
  FEES, IF ANY,
  THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THIS SOFTWARE.

  MICROCHIP PROVIDES THIS SOFTWARE CONDITIONALLY UPON YOUR
  ACCEPTANCE OF THESE
  TERMS.
*/
/**
```

```

    Section: Included Files
*/

#define FOSC 120000000
#define FCY FOSC/2

#include "mcc_generated_files/system.h"
#include "adc1.h"
#include "pin_manager.h"
#include "libpic30.h"
#include "pwm.h"
#include "Variables.h"
#include "dsp.h"
#include "xc.h"
#include "stdlib.h"
#include "stdio.h"
#include "math.h"
#include "CompensadorDigital.h"
#include "smps_control.h"
#include <stdint.h>
#include "p33EV256GM002.h"

// Punteros para el historial del controlador
int controlador2P2ZControlHistory[2] = {0,0};
int controlador2P2ZErrorHistory[3] = {0,0,0};

// Declaramos los coeficientes del controlador

int A1 = Q15(-1.242/8);    //A1
int A2 = Q15(0.2417/8);   //A2
int B0 = Q15(5.415/8);    //B0
int B1 = Q15(1.026/8);    //B1
int B2 = Q15(-4.389/8);   //B2

int SobreCorriente = 0;

/*
                                Main application
*/
int main(void)
{
    // initialize the device
    SYSTEM_Initialize();

    // Se inicializan los punteros para los datos del controlador
    controlador2P2ZControlHistory[0] = 0;
    controlador2P2ZControlHistory[1] = 0;
    controlador2P2ZErrorHistory[0] = 0;
    controlador2P2ZErrorHistory[1] = 0;
    controlador2P2ZErrorHistory[2] = 0;

// ----- // Bucle principal del programa // -----
// ----- //

    while (1)
    {
        // ----- // Lectura Voltaje Secundario // -----
// ----- //
        ADC1_ChannelSelectSet(0x0);

```

```

ADC1_SamplingStart();
ADC1_SamplingStop();
while(!ADC1_IsConversionComplete())
{
    ADC1_Tasks();
}
VoltajeSec =ADC1_Channel0ConversionResultGet() ;

// ----- // Lectura Corriente Secundario // -----
----- //

ADC1_ChannelSelectSet(0x1);
ADC1_SamplingStart();
ADC1_SamplingStop();
while(!ADC1_IsConversionComplete())
{
    ADC1_Tasks();
}
CorrienteSec =ADC1_Channel0ConversionResultGet() ;

// ----- // Lectura Corriente Primario // -----
----- //

ADC1_ChannelSelectSet(0x2);
ADC1_SamplingStart();
ADC1_SamplingStop();
while(!ADC1_IsConversionComplete())
{
    ADC1_Tasks();
}
CorrientePrim =ADC1_Channel0ConversionResultGet() ;

// Escalamos algunas medidas a Q15. Formato mucho más rápido
que float que nos permite aprovechar las funciones DSC.
// Como los multiplicadores Hardware. Al pasar los valores a
Q15, se escalan a un fondo de 32768.
// Al tener un microcontrolador de 16 bits, nos deja un bit de
signo y margen para overflow

// Escalamos los valores del ADC al fondo de 32768 = 4096 * 8
(2^3)
VSecQ12 = VoltajeSec;
ISecQ12 = CorrienteSec;
IPrimQ15 = (CorrientePrim << 3); // Se escala a 32768

// Comprobación corriente primario
if (CorrientePrim > 3277)
{
    PWM_ModuleDisable(); // Si la corriente es mayor
de 3.72A, apaga el módulo PWM
    SobreCorriente = 1;
}
else
{
    PWM_ModuleEnable();
    SobreCorriente = 0;
}

// ----- // COMPENSADOR DIGITAL // -----
//
    
```

```

// ----- // CALCULO LAZO DE CONTROL // -----
----- //

// Calculamos el error del lazo de tensión. Valor máximo del
error = VSecError = VConsigna - 0V (Leidos)
VSecError = VoltajeConsignaQ12 - VSecQ12;

// Guardamos el error en el registro del controlador
controlador2P2ZErrorHistory[2] =
controlador2P2ZErrorHistory[1]; // e(n-2)
controlador2P2ZErrorHistory[1] =
controlador2P2ZErrorHistory[0]; // e(n-1)
controlador2P2ZErrorHistory[0] = VSecError;
// e(n))

// Actualizamos el registro de acciones de control
controlador2P2ZControlHistory[1] =
controlador2P2ZControlHistory[0]; // u(n-2)
controlador2P2ZControlHistory[0] = SalidaControlador;
// u(n-1)

// Controlador 2p2z

SalidaControlador = B0*controlador2P2ZErrorHistory[0] +
B1*controlador2P2ZErrorHistory[1] + B2*controlador2P2ZErrorHistory[2]
+ A1*controlador2P2ZControlHistory[0] +
A2*controlador2P2ZControlHistory[1];

// Dado que los coeficientes se han dividido por 8
SalidaControlador = SalidaControlador*8;

if (SalidaControlador > 32768)
    SalidaControlador = 32768;
else if (SalidaControlador < -32768)
    SalidaControlador = -32768;

// Rampa moduladora digital
IPrimRefQ15 = SalidaControlador + DeltaRamp;

// Comparador de corriente
DeltaUT = IPrimQ15 - IPrimRefQ15;

// Saturación
if (DeltaUT > 32768)
    DeltaUT = 32768;
else if (DeltaUT < -32768)
    DeltaUT = -32768;

// Ahora ajustamos la salida del controlador a la accion de
control ut, que puede valer como máximo 600d para tener un DT del 50%
// Podriamos escalar DeltaPID a 512. Perdemos un poco de fase,
pero evitamos entrar en hard switching.
// Esto nos deja un DT máximo en el TF: 42.66 % ES MUY POCO
// PARA TENER UN 48% DE DT TENDRIA QUE VALER 576 MAXIMO =
32768 ENTRE 57.
// O lo que es lo mismo 32768*1,125/64 = 576
// Como no podemos dividir por un numero que no sea múltiplo
de 2. Primero dividimos DeltaPID por 2^3 que nos da el 12,5% de lo que
valga

```

```
// Luego se lo sumamos para obtener el resultado de
multiplicar por 1.125. Y por ultimo, lo dividimos entre 64

CalculoDocePorCiento = (DeltaUT >> 3);
DeltaUT = DeltaUT + CalculoDocePorCiento;
DeltaUT = (DeltaUT >> 6); // Se divide por 64

// Variación de la fase PWM según la salida del lazo de
control
ut = FaseAnterior + DeltaUT;
FaseAnterior = ut;

if (ISecQ12 > CorrienteConsignaQ12)
{
    CCCV = 1;
    FaseLimite = ut; // Limitar fase para que no suba
    ut = FaseLimite;
}

//Anti-windup general
if (ut > FaseMax)
{
    ut = FaseMax;
    FaseAnterior = ut;
}
else if (ut < FaseMin)
{
    ut = -ut;
    if (ut > FaseMax)
    {
        ut = FaseMax;
        FaseAnterior = ut;
    }
}

// Cambiar el desfase PWM 0 - 600
PWM_PeriodCenterAlignedModeSet(PWM_GENERATOR_2, ut);

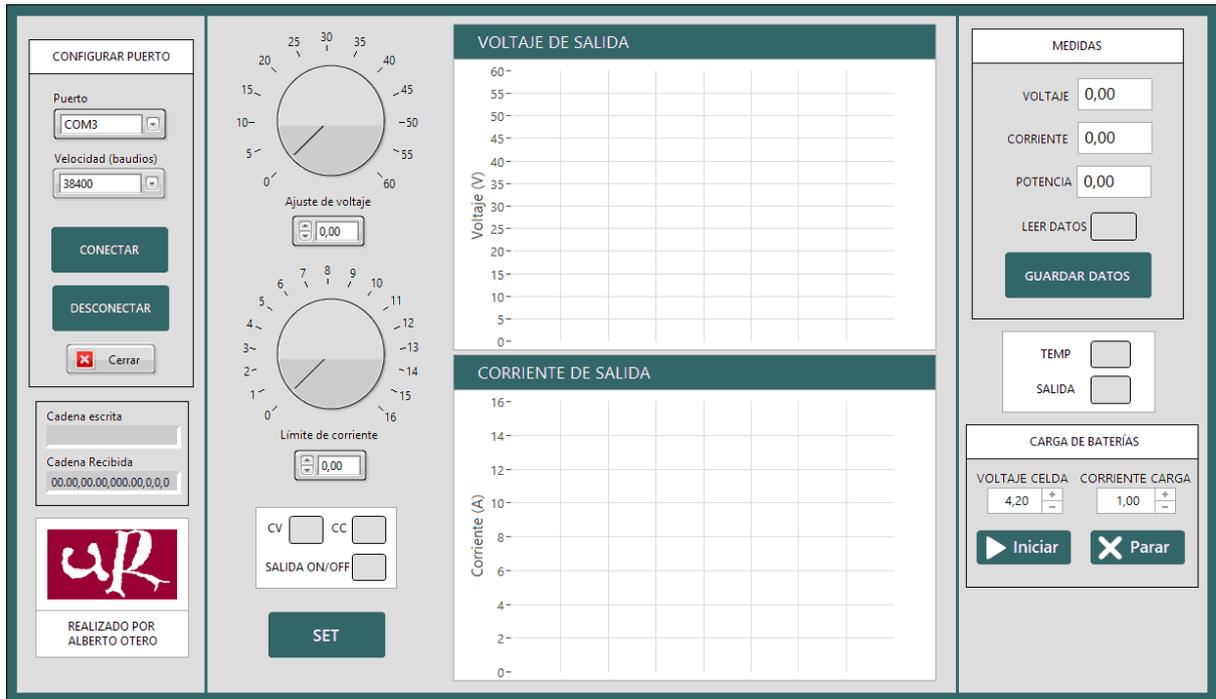
if (OUT == 1 && Temp == 0) {
    // Se enciende relé de salida
    PORTBbits.RB8 = 1;
    // Lazo de control PID
    //CompensadorDigital();
}
else if (OUT == 1 && Temp == 1) {
    // Relé de salida apagado
    PORTBbits.RB8 = 0;
}
}

/**
End of File
*/
```

## 4. Programa Control remoto fuente de alimentación LabVIEW

Se presenta a continuación, el programa en LabVIEW que controla el puerto USB a través de la librería VISA, y que permite modificar las consignas de tensión y corriente, además de iniciar o para la carga de una batería. Se adjunta, además, las rutinas que lo forman.

### 4.1. Interfaz de usuario



La interfaz de usuario, consiste en un panel donde se pueden configurar las comunicaciones, las consignas, así como monitorizar los parámetros de la salida y las cadenas recibidas y enviadas para depuración.

A la izquierda se encuentra la configuración del puerto para conectarse a la fuente, cuya velocidad debe ser de 38400 baudios. Una vez se ha pulsado conectar, la fuente queda a la espera, con la salida desactivada, de recibir la consigna del usuario, que se establecerá con el botón SET.

El usuario puede configurar tensión y corriente y activar la salida. Además, puede establecer la tensión de carga de una batería y la corriente a la que quiere cargarla con el botón Iniciar carga. Una vez se ha finalizado el proceso de carga, aparece el mensaje “Carga completada”. El usuario también puede detener la carga en todo momento con el botón Parar carga.

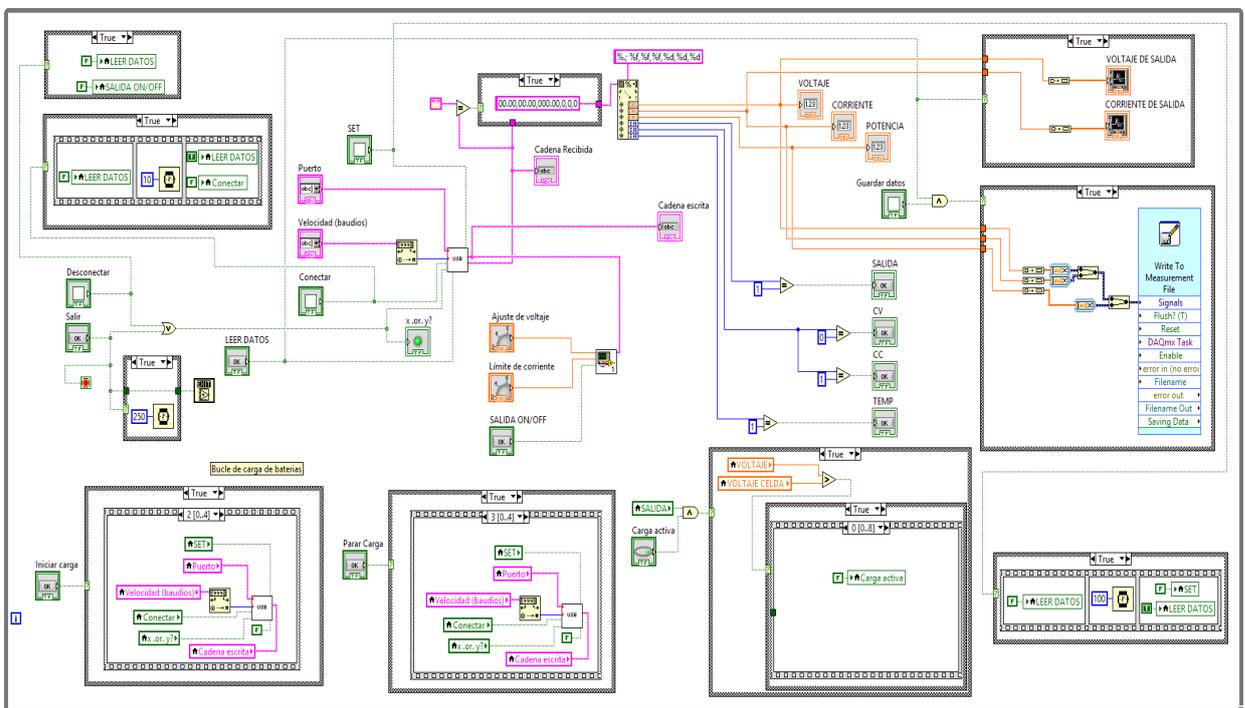
Puede guardar los resultados en un archivo Excel a través del botón Guardar Datos.

## 4.2. Variables locales

-  **Salir**
-  **Velocidad (baudios) baud rate is the rate of transmission.**
-  **Puerto**
-  **Conectar [Conectar ]**
-  **Desconectar[Desconectar]**
-  **Guardar datos**
-  **Iniciar carga**
-  **Parar Carga**
-  **SALIDA ON/OFF**
-  **LEER DATOS**
-  **SET**
-  **VOLTAJE CELDA**
-  **CORRIENTE CARGA**
-  **Ajuste de voltaje**
-  **Límite de corriente**
-  **Carga activa**
-  **VOLTAJE**
-  **CORRIENTE**
-  **POTENCIA**
-  **SALIDA**
-  **TEMP**
-  **Cadena Recibida**
-  **Cadena escrita**
-  **Waveform Chart - dark[VOLTAJE DE SALIDA]**

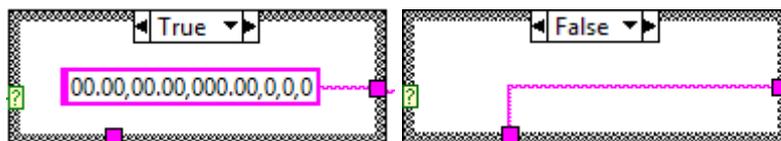
-  Waveform Chart - dark[**CORRIENTE DE SALIDA**]
-  CV
-  CC
-  x .or. y?

### 4.3. Diagrama de bloques

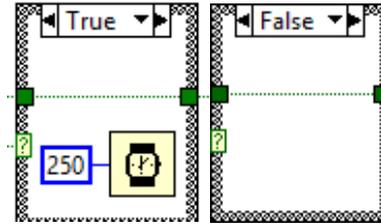


### 4.4. Bucles

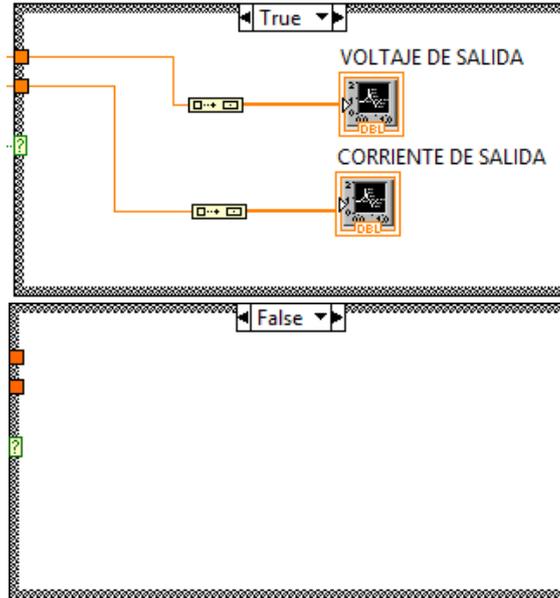
Cuando no se ha recibido ninguna cadena todavía, para evitar que el convertor de cadenas muestre un error, se inicia con una cadena en blanco. Si se recibe una cadena, se lee la que viene por el puerto



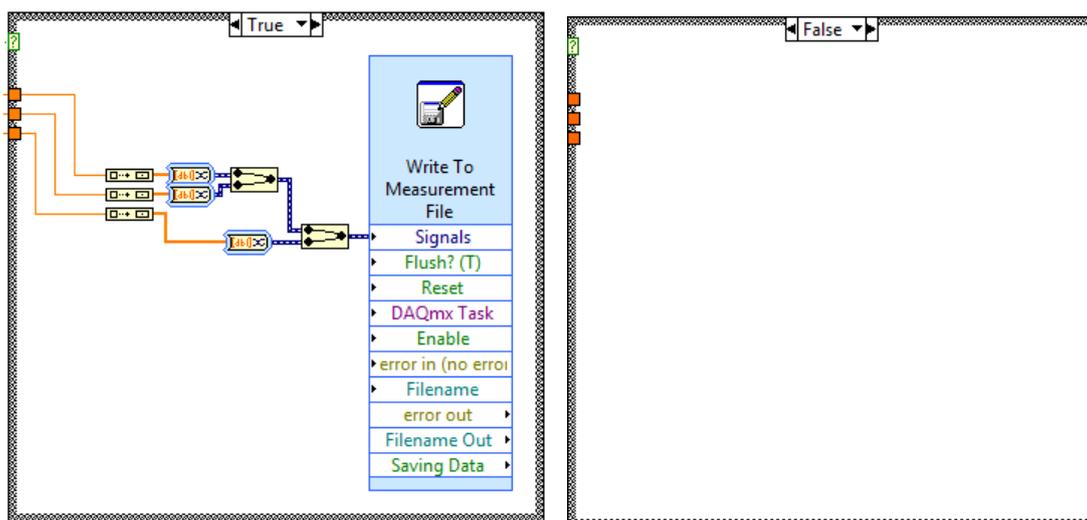
Si se pulsa Salir, se cierra la aplicación. Si no, se queda a la espera



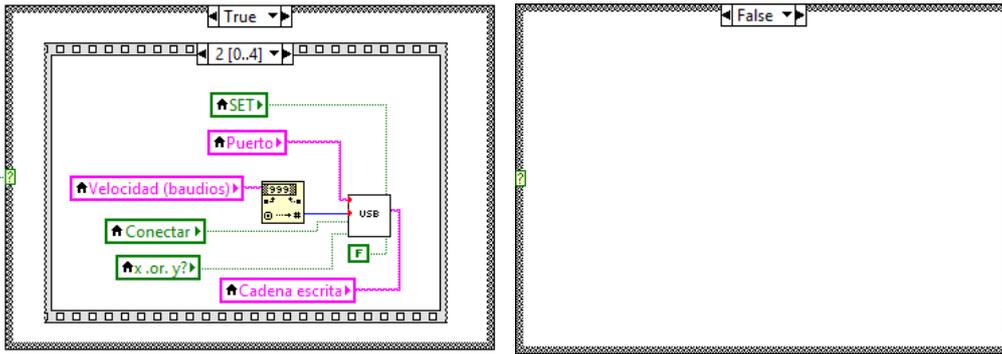
Si Leer Datos está activo, se muestran los valores en las gráficas. Si no, se desactivan.



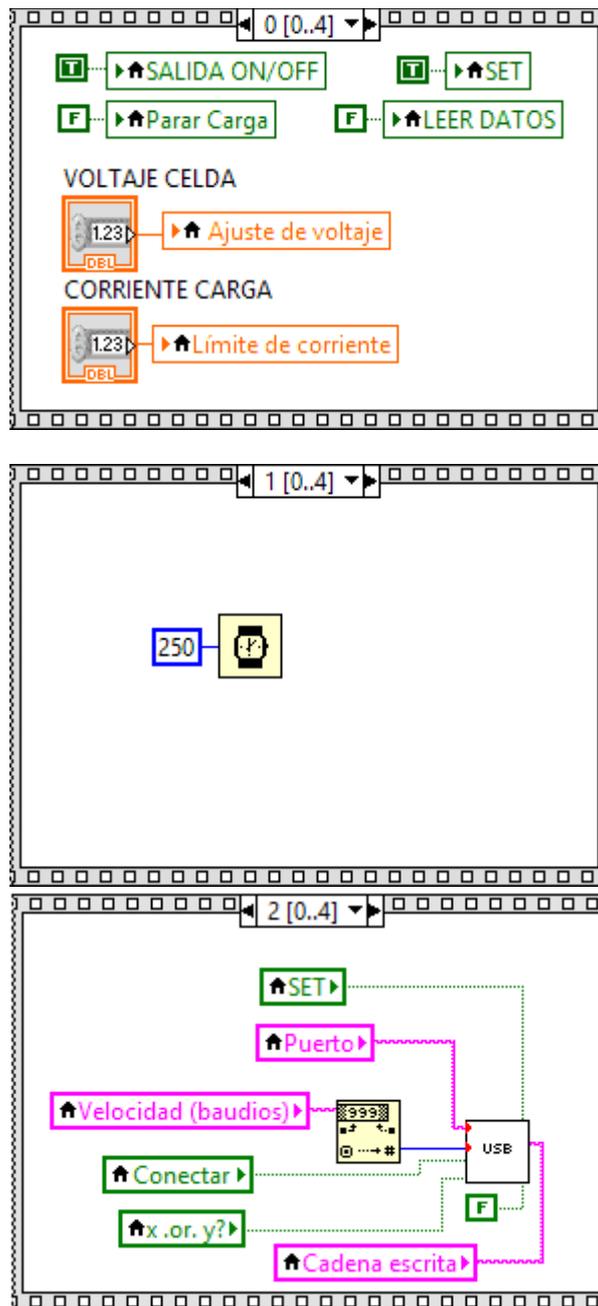
Si Guardar Datos esta activo, se genera un archivo Excel con los valores de voltaje, corriente y potencia, si no, este módulo no está activo.

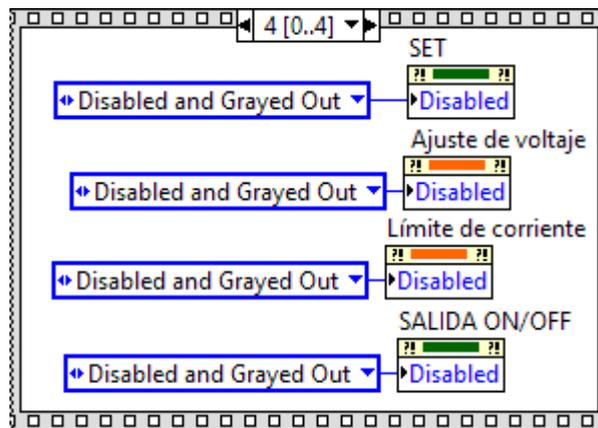
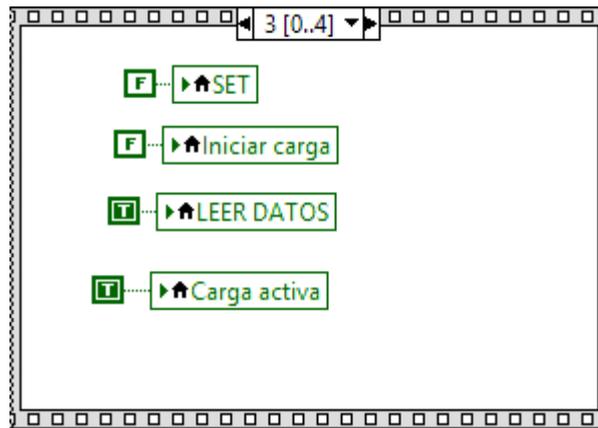


Para iniciar la carga de baterías, el control Iniciar Carga debe estar activo, y se ejecuta el siguiente bucle:

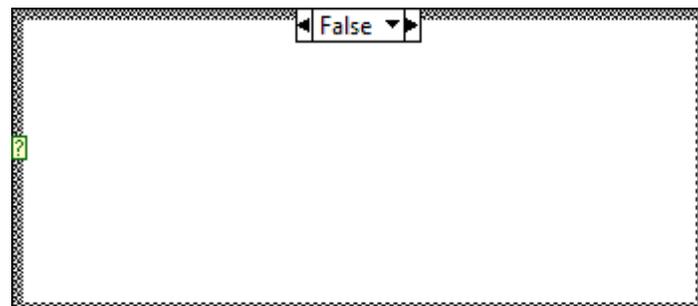
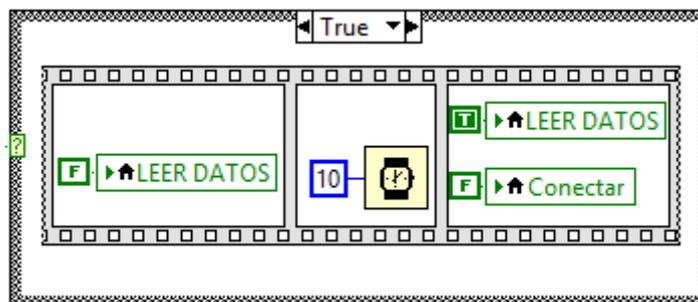


La secuencia de activación de la carga es la siguiente:





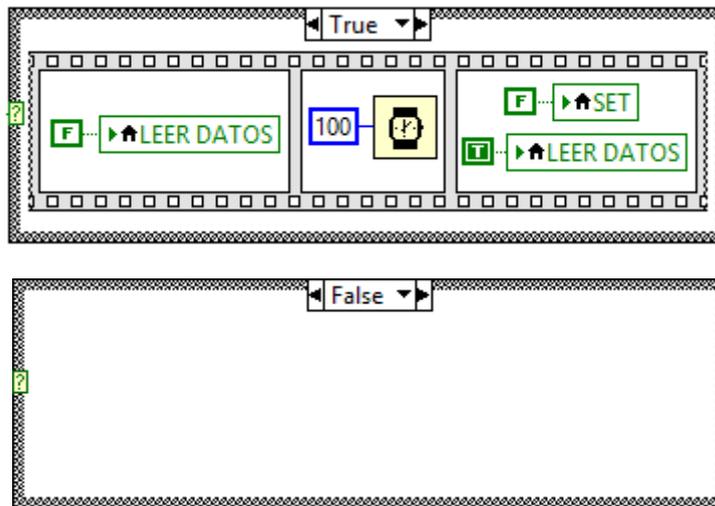
Para conectar una con la fuente, se debe pulsar Conectar, y se ejecuta el siguiente bucle:



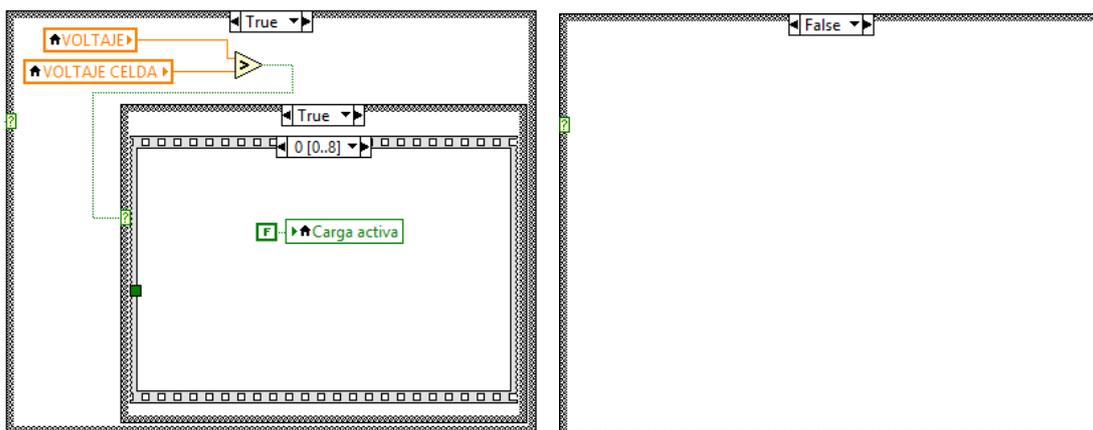
Pulsar Desconectar, deshabilita el control Leer Datos y desactiva la salida:



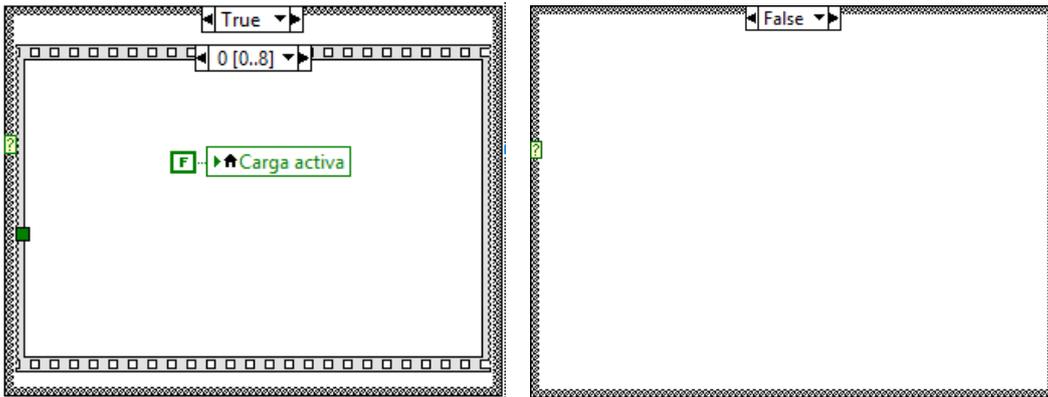
Para enviar una consigna a la fuente, se debe pulsar SET y se ejecuta el siguiente bucle:



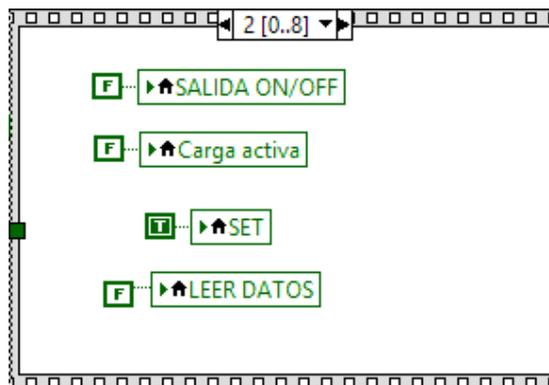
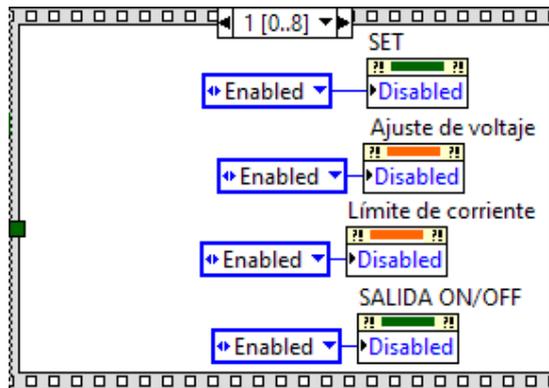
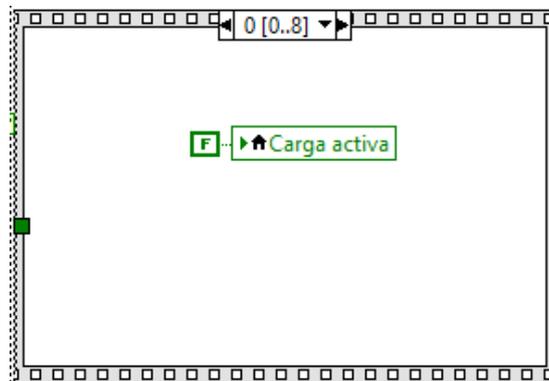
Si el indicador Carga Activa, está a uno, se evalúa continuamente si se ha llegado al voltaje de consigna. Si esto es así se ejecuta la subrutina que detiene la carga. Si Carga Activa está desactiva no se evalúa este bucle.

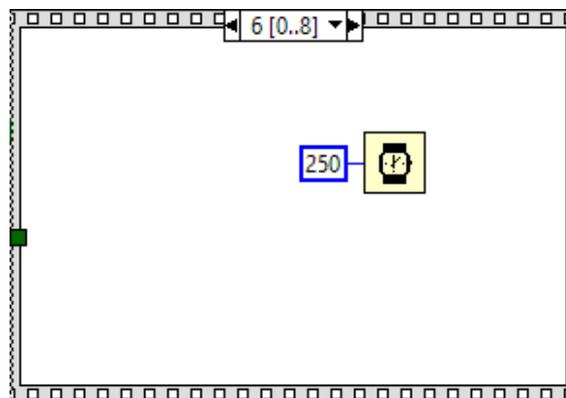
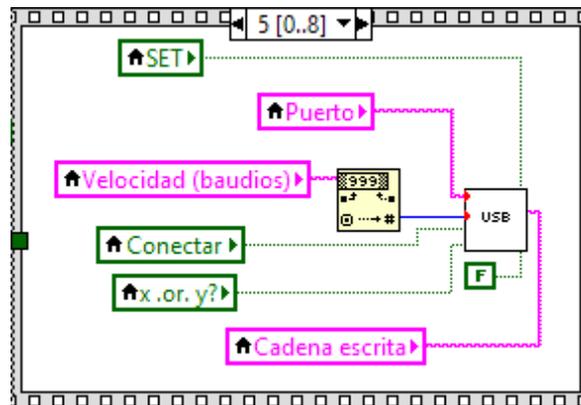
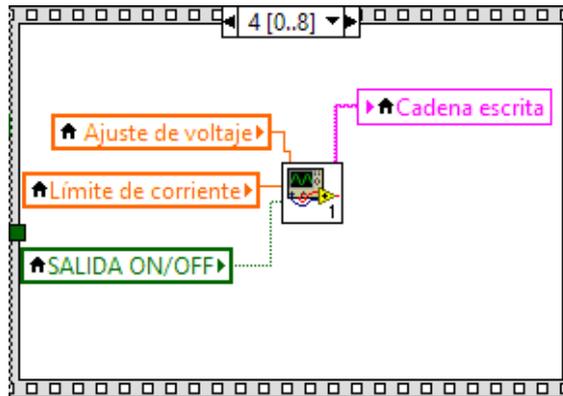
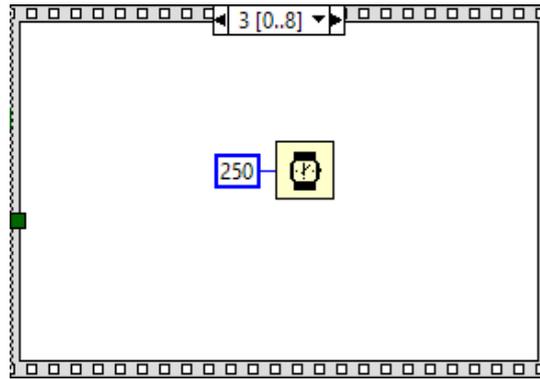


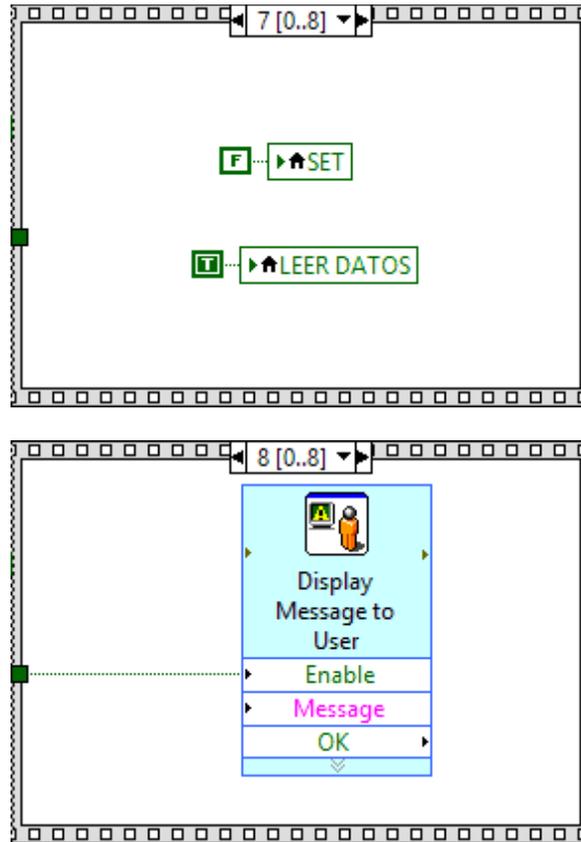
Mientras que no se active el comparador, sigue cargando.



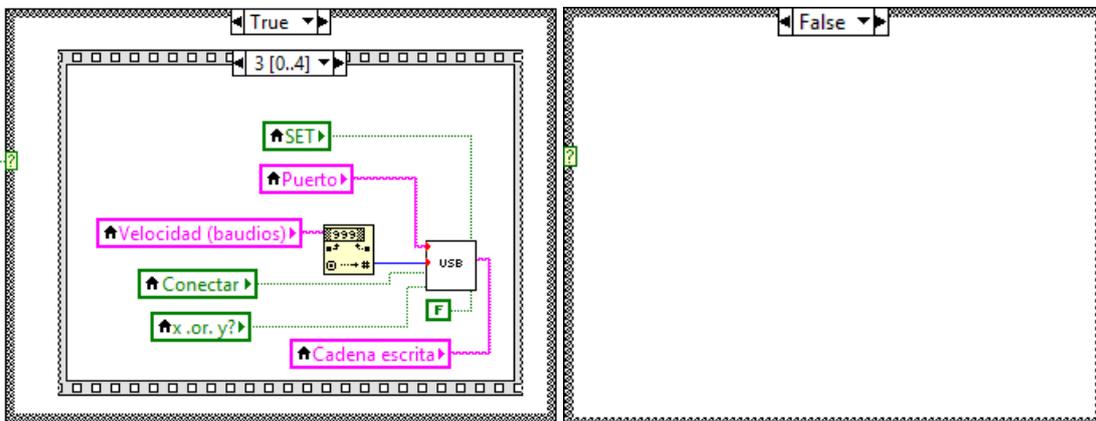
La secuencia de desactivación es la siguiente:



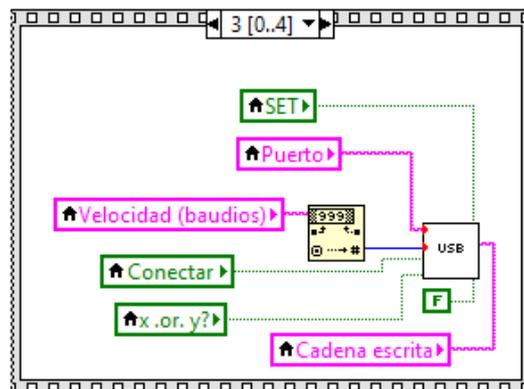
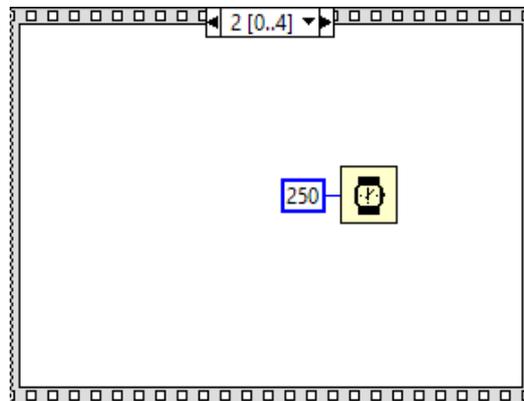
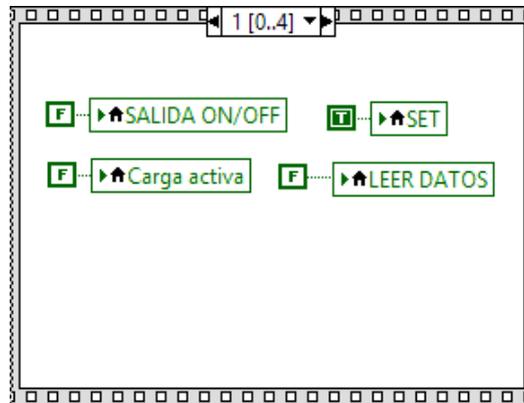
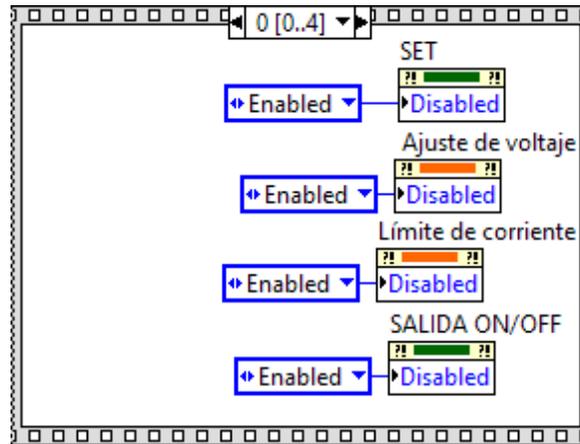


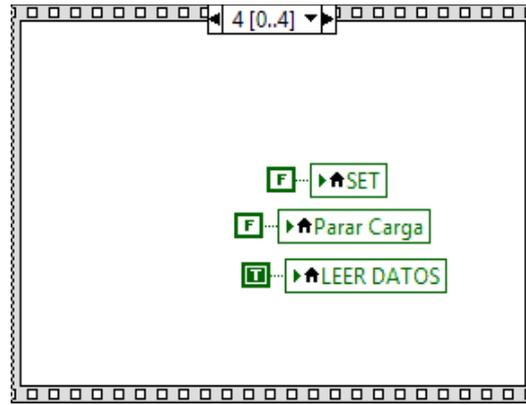


También se puede detener la carga de forma manual, pulsando Parar Carga:



La secuencia de desactivación es la siguiente:





## 4.5. Dependencias



### subDisplayMessage.vi

C:\Program Files\National Instruments\LabVIEW 2015\vi.lib\express\express output\DisplayMessageBlock.llb\subDisplayMessage.vi



### ex\_subExcelFileWrite.vi

C:\Program Files\National Instruments\LabVIEW 2015\vi.lib\express\express output\ExFileWriteBlock.llb\ex\_subExcelFileWrite.vi



### ex\_NewFileOptions.ctl

C:\Program Files\National Instruments\LabVIEW 2015\vi.lib\express\express output\ExFileWriteBlock.llb\ex\_NewFileOptions.ctl



### ex\_propertySource.ctl

C:\Program Files\National Instruments\LabVIEW 2015\vi.lib\express\express shared\ExFile.llb\ex\_propertySource.ctl



### usiDataType.ctl

C:\Program Files\National Instruments\LabVIEW 2015\vi.lib\express\express shared\ex\_EditUserDefinedProperties\usiDataType.ctl



### ex\_userDefProperty.ctl

C:\Program Files\National Instruments\LabVIEW 2015\vi.lib\express\express shared\ExFile.llb\ex\_userDefProperty.ctl



### ex\_ExcelAdvancedConfig.ctl

C:\Program Files\National Instruments\LabVIEW 2015\vi.lib\express\express output\ExFileWriteBlock.llb\ex\_ExcelAdvancedConfig.ctl



### ex\_FileFormats.ctl

C:\Program Files\National Instruments\LabVIEW 2015\vi.lib\express\express shared\ExFile.llb\ex\_FileFormats.ctl



### FormarCadena.vi

C:\Users\Alberto\Google Drive\TFG Alberto Otero Olavarrieta\TFG Fuente de alimentación\Código\Programa LabView - Terminado\Programa Completo\FormarCadena.vi



### Waveform Array To Dynamic.vi

C:\Program Files\National Instruments\LabVIEW 2015\vi.lib\express\express shared\transition.lib\Waveform Array To Dynamic.vi

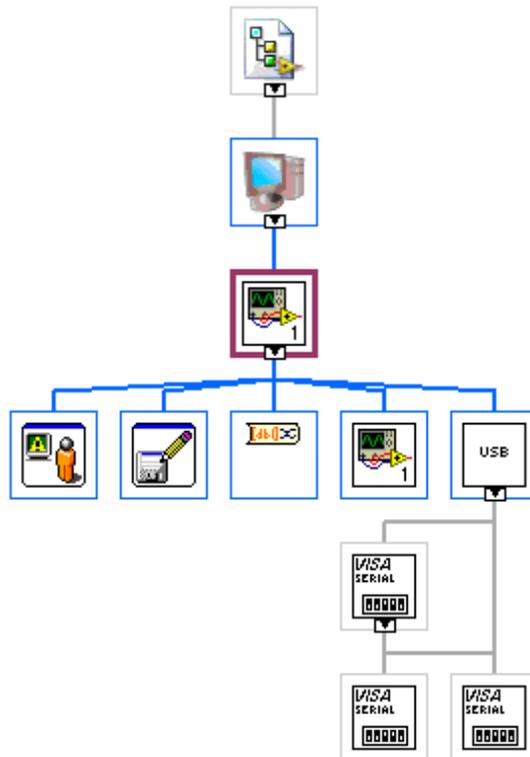


**Prueba comunicaciones.vi**

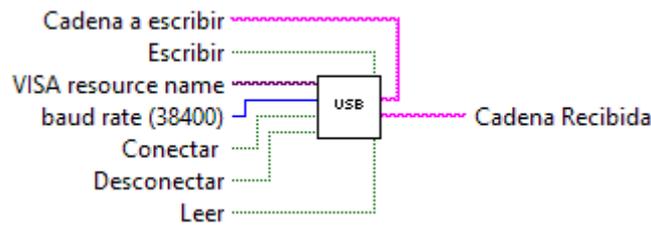
C:\Users\Alberto\Google Drive\TFG Alberto Otero Olavarrieta\TFG Fuente de alimentación\Código\Programa LabView - Terminado\Comunicación Serie USB\Prueba comunicaciones.vi

"Control Remoto Fuente de alimentación.vi History"  
Current Revision: 122

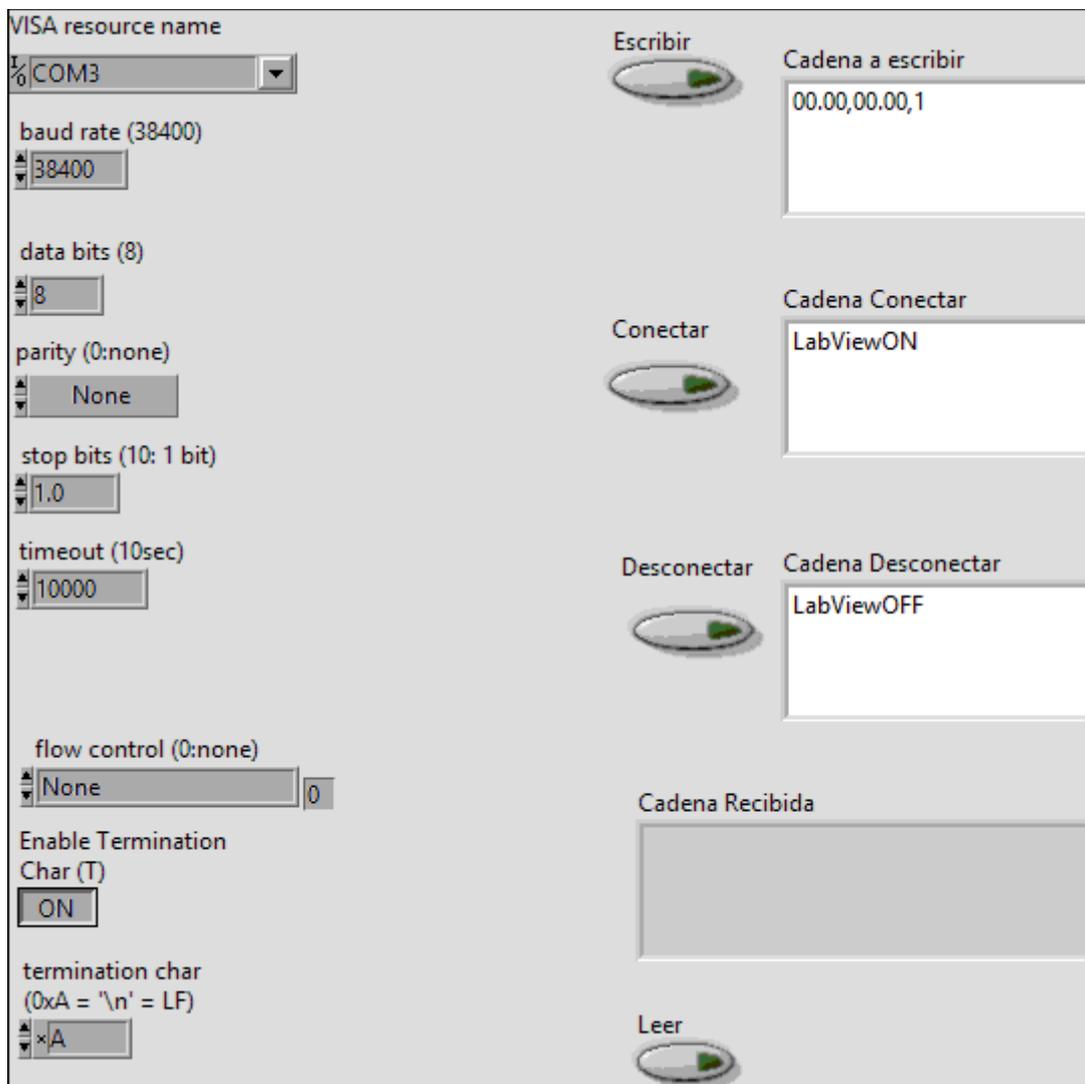
4.6. Jerarquía



### 4.7. Programa LabVIEW – Prueba comunicaciones.vi



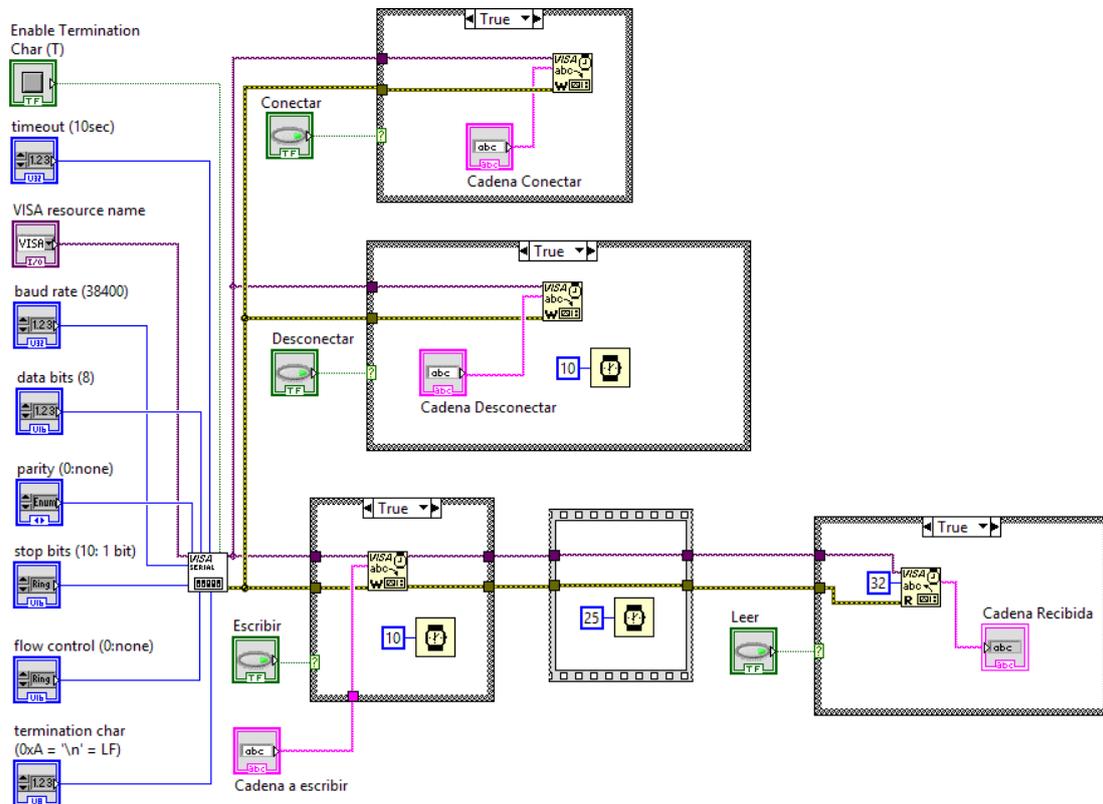
### 4.8. Interfaz de usuario



### 4.9. Variables locales

- abc Cadena a escribir
- TF Escribir
- I/O **VISA resource name** VISA resource name specifies the resource to be opened. The VISA resource name control also specifies the session and class.
- U32 **baud rate (38400)** baud rate is the rate of transmission.
- TF Conectar
- TF Desconectar
- TF Leer
- abc Cadena Recibida

### 4.10. Diagrama de bloques

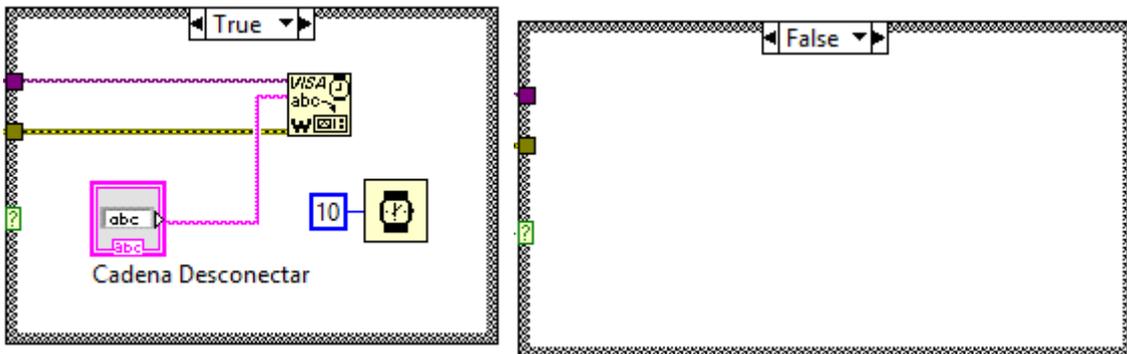


### 4.11. Bucles

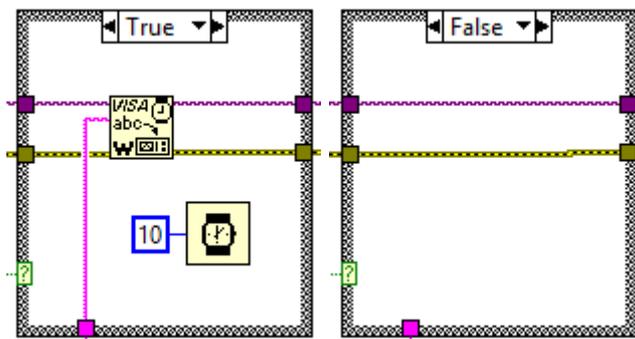
Cuando el control Conectar se activa, se envía “Cadena Conectar” por el puerto, que contiene “LabViewON”. Si no esta activo no envía nada.



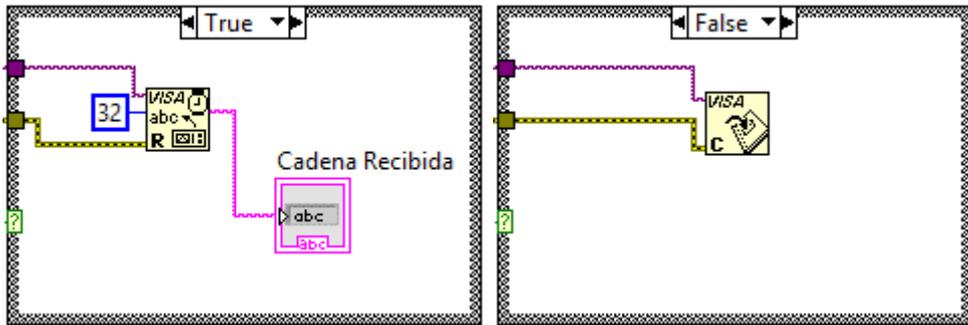
Cuando el control Desconectar se activa, se envía “Cadena Desconectar” por el puerto, que contiene “LabViewOFF”. Si no está activo no envía nada.



Cuando se activa el control Escribir, se envía por el puerto “Cadena a escribir”. Si no, no se envía nada y se pasa a la recepción.



Si el control Leer está activo, se recibe lo que el microcontralador ATmega328 envía por el puerto en “Cadena Recibida”. Si no, se cierra el puerto.



### 4.12. Dependencias



**VISA Configure Serial Port (Instr).vi**

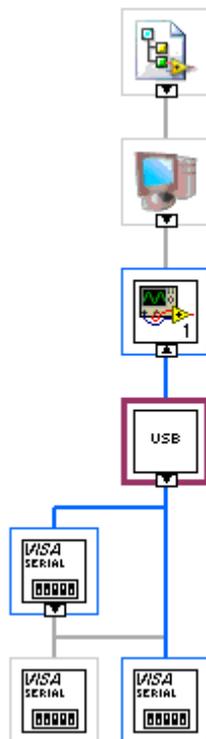
C:\Program Files\National Instruments\LabVIEW 2015\vi.lib\Instr\\_visa.lib\VISA Configure Serial Port (Instr).vi



**VISA Configure Serial Port**

C:\Program Files\National Instruments\LabVIEW 2015\vi.lib\Instr\\_visa.lib\VISA Configure Serial Port

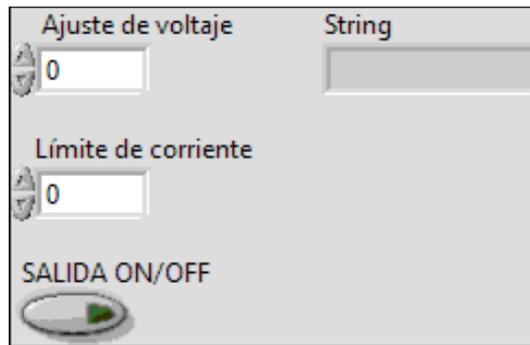
### 4.13. Jerarquía



### 4.14. Programa LabVIEW – Formar Cadena.vi

Este subprograma se encarga de formar la cadena antes de ser escrita en el puerto en los bucles de iniciar y parar carga.

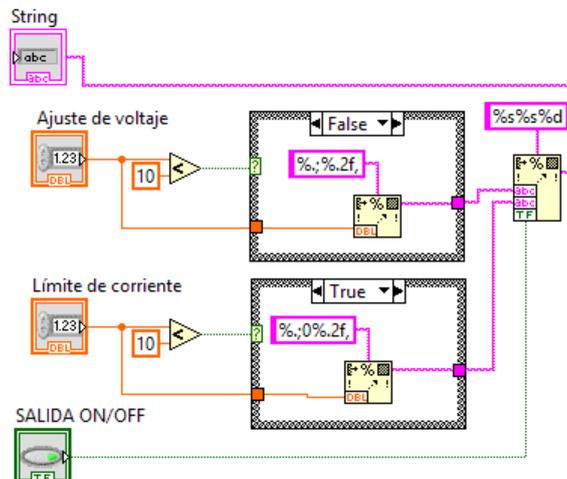
### 4.15. Interfaz de usuario



### 4.16. Variables locales

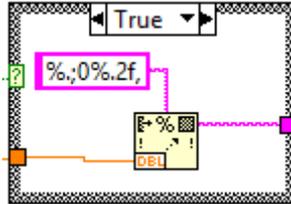
-  **Límite de corriente**
-  **Ajuste de voltaje**
-  **SALIDA ON/OFF**
-  **String**

### 4.17. Diagrama de bloques

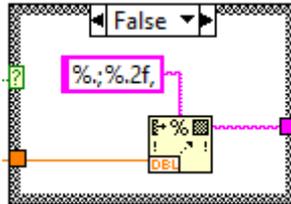


### 4.18. Bucles

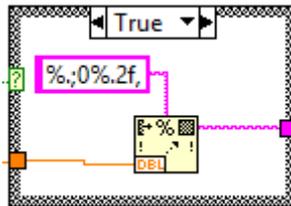
El bucle de Ajuste de tensión cuando la consigna es menor que 10, añade un 0 a la posición de la cadena



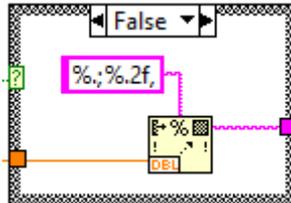
Cuando es falso, no la modifica



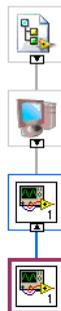
El bucle de Ajuste de corriente cuando la consigna es menor que 10, añade un 0 a la posición de la cadena



Cuando es falso, no la modifica



### 4.19. Jerarquía



# PLANOS



## Índice de planos

### Planos mecánicos

1. Fuente completa
2. Paneles de plástico. Panel Frontal – Vista exterior
3. Paneles de plástico. Panel Frontal – Vista interior
4. Paneles de plástico. Panel Trasero
5. Carcasa exterior – Lámina de acero inferior
6. Carcasa exterior – Lámina de acero superior

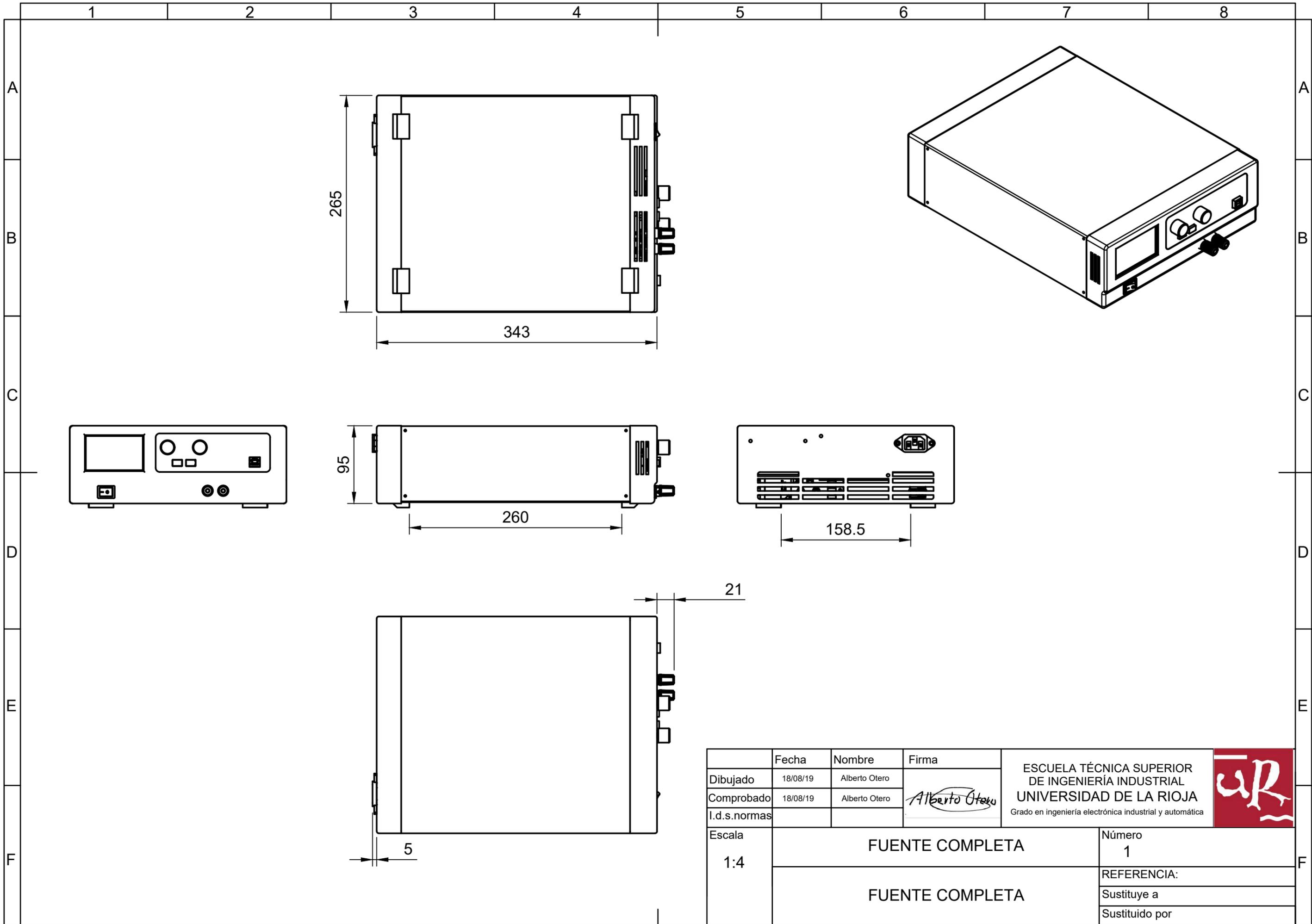
### Planos eléctricos

1. Controlador PFC. Esquema eléctrico
2. Controlador PFC. Layout Componentes
3. Controlador PFC. Layout Agujeros
4. Controlador PFC. Pistas – Capa inferior
5. Controlador PFC. Pistas – Capa superior
6. Etapa de entrada y filtro EMI. Esquema eléctrico
7. Etapa de entrada y filtro EMI. Layout Componentes
8. Etapa de entrada y filtro EMI. Layout Agujeros
9. Etapa de entrada y filtro EMI. Pistas – Capa inferior
10. Convertidor Boost PFC. Esquema eléctrico
11. Convertidor Boost PFC. Layout Componentes
12. Convertidor Boost PFC. Layout Agujeros
13. Convertidor Boost PFC. Pistas – Capa inferior
14. Convertidor Boost PFC. Pistas – Capa superior
15. Convertidor DC – DC ZVS. Esquema eléctrico
16. Convertidor DC – DC ZVS. Layout Componentes
17. Convertidor DC – DC ZVS. Layout Agujeros
18. Convertidor DC – DC ZVS. Pistas – Capa inferior
19. Convertidor DC – DC ZVS. Pistas – Capa superior
20. Controlador DC – DC ZVS. Esquema eléctrico
21. Controlador DC – DC ZVS. Layout Componentes
22. Controlador DC – DC ZVS. Layout Agujeros
23. Controlador DC – DC ZVS. Pistas – Capa inferior
24. Controlador DC – DC ZVS. Pistas – Capa superior
25. Botones y Puerto USB. Esquema eléctrico
26. Botones y Puerto USB. Layout Componentes
27. Botones y Puerto USB. Layout Agujeros
28. Botones y Puerto USB. Pistas – Capa inferior

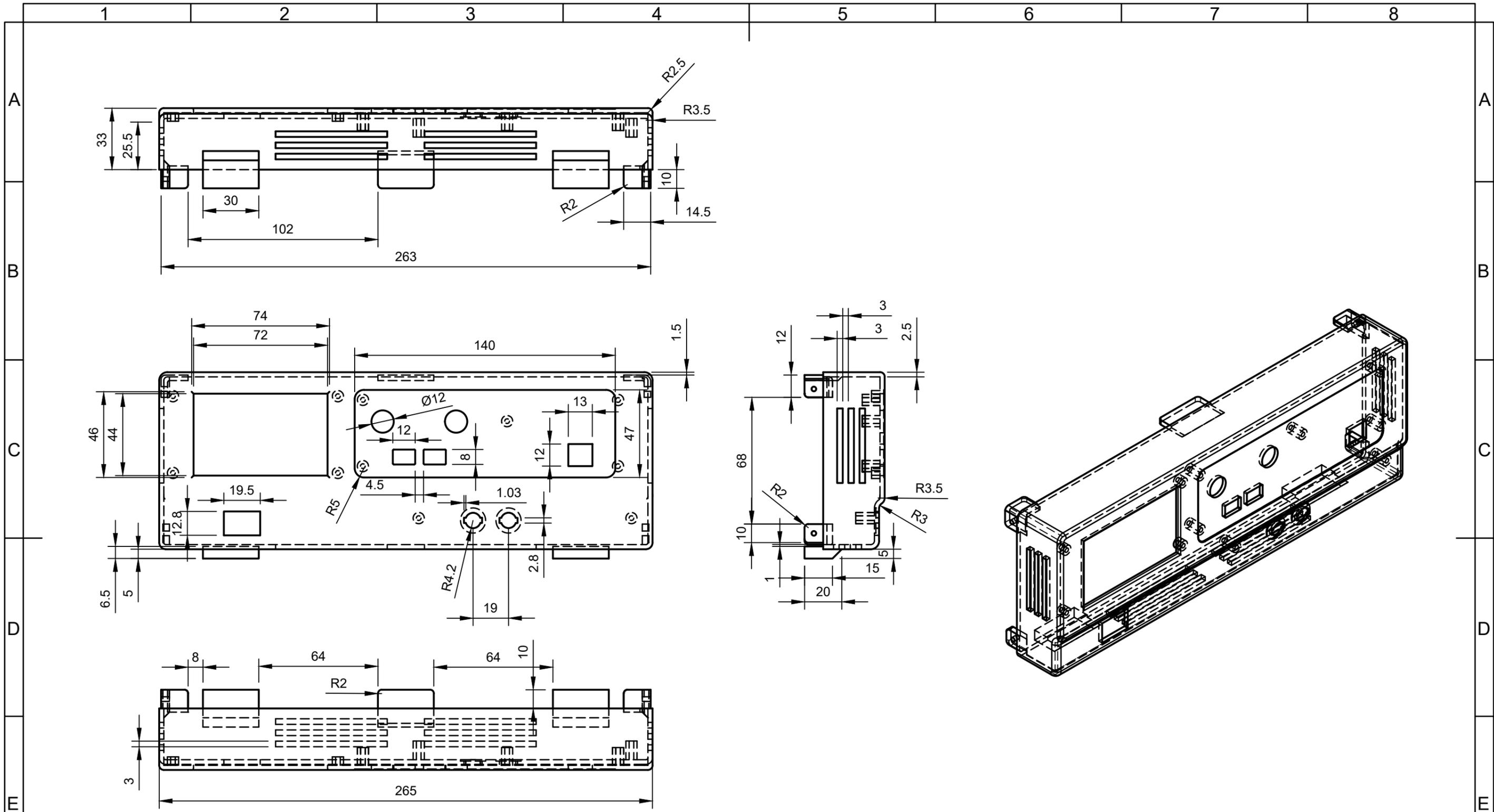
29. Botones y Puerto USB. Pistas – Capa superior
30. Controlador panel frontal. Esquema eléctrico
31. Controlador panel frontal. Layout Componentes
32. Controlador panel frontal. Layout Agujeros
33. Controlador panel frontal. Pistas – Capa inferior
34. Controlador panel frontal. Pistas – Capa superior
35. Controlador de temperatura. Esquema eléctrico
36. Controlador de temperatura. Layout Componentes
37. Controlador de temperatura. Layout Agujeros
38. Controlador de temperatura. Pistas – Capa inferior
39. Controlador de temperatura. Pistas – Capa superior
40. Driver puente completo. Esquema eléctrico
41. Driver puente completo. Layout Componentes
42. Driver puente completo. Layout Agujeros
43. Driver puente completo. Pistas – Capa inferior
44. Driver rectificación síncrona. Esquema eléctrico
45. Driver rectificación síncrona. Layout Componentes
46. Driver rectificación síncrona. Layout Agujeros
47. Driver rectificación síncrona. Pistas – Capa inferior
48. Etapa de salida. Esquema eléctrico
49. Etapa de salida. Layout Componentes
50. Etapa de salida. Layout Agujeros
51. Etapa de salida. Pistas – Capa inferior

# PLANOS MECÁNICOS

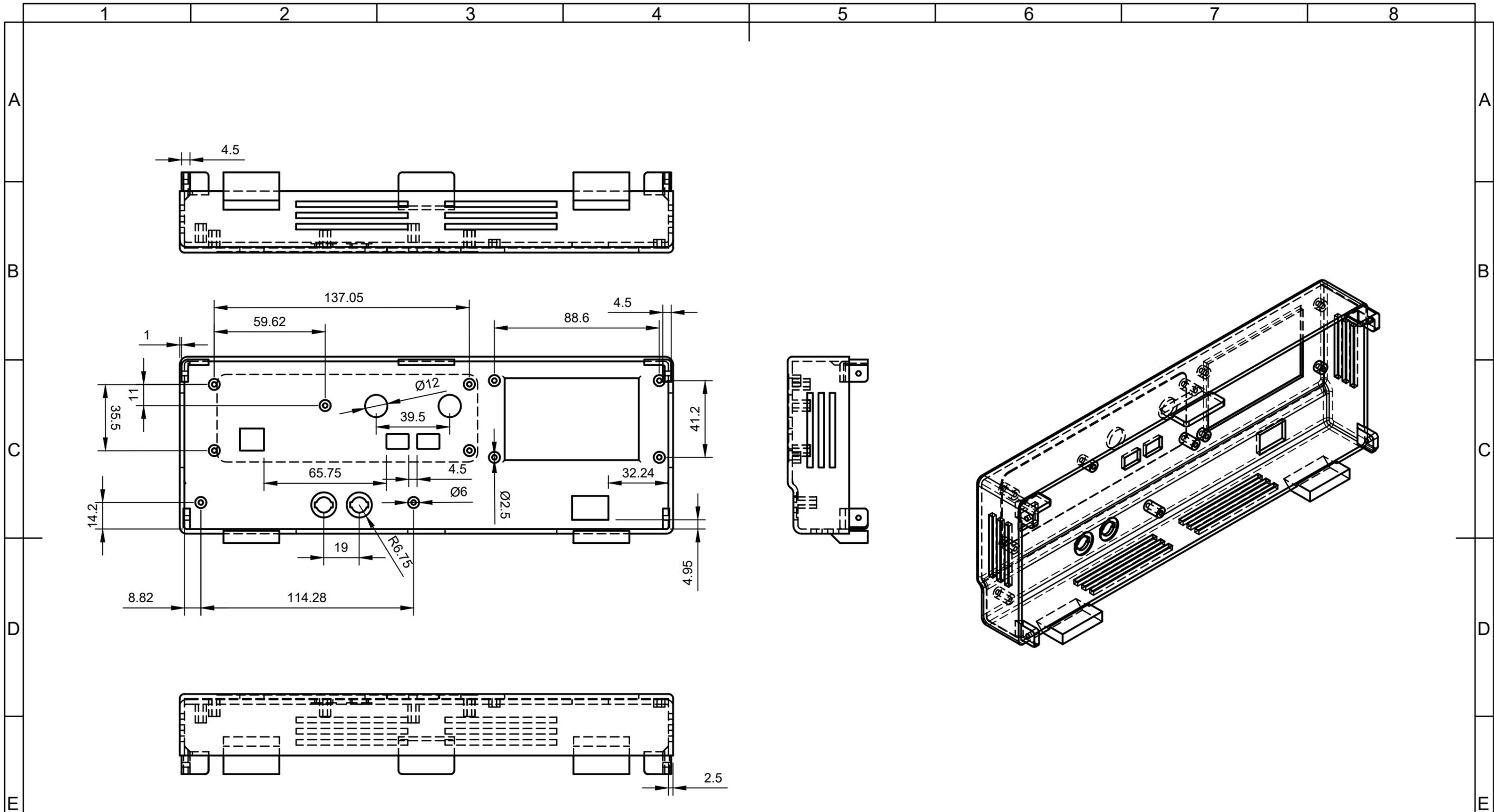




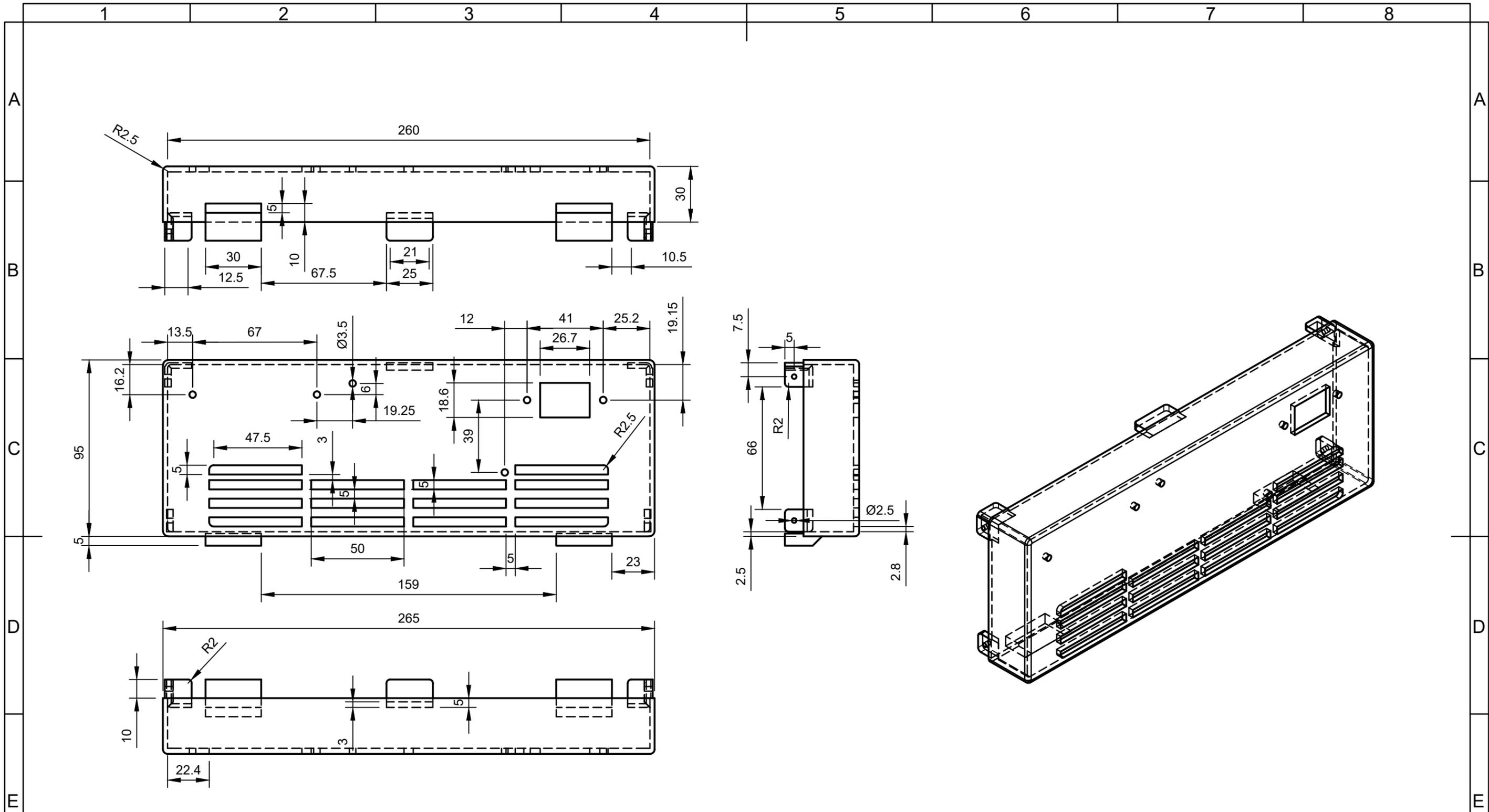
	Fecha	Nombre	Firma	ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL UNIVERSIDAD DE LA RIOJA <small>Grado en ingeniería electrónica industrial y automática</small> 
Dibujado	18/08/19	Alberto Otero		
Comprobado	18/08/19	Alberto Otero		
I.d.s.normas				
Escala	FUENTE COMPLETA		Número	
1:4	FUENTE COMPLETA		1	
			REFERENCIA:	
			Sustituye a	
			Sustituido por	



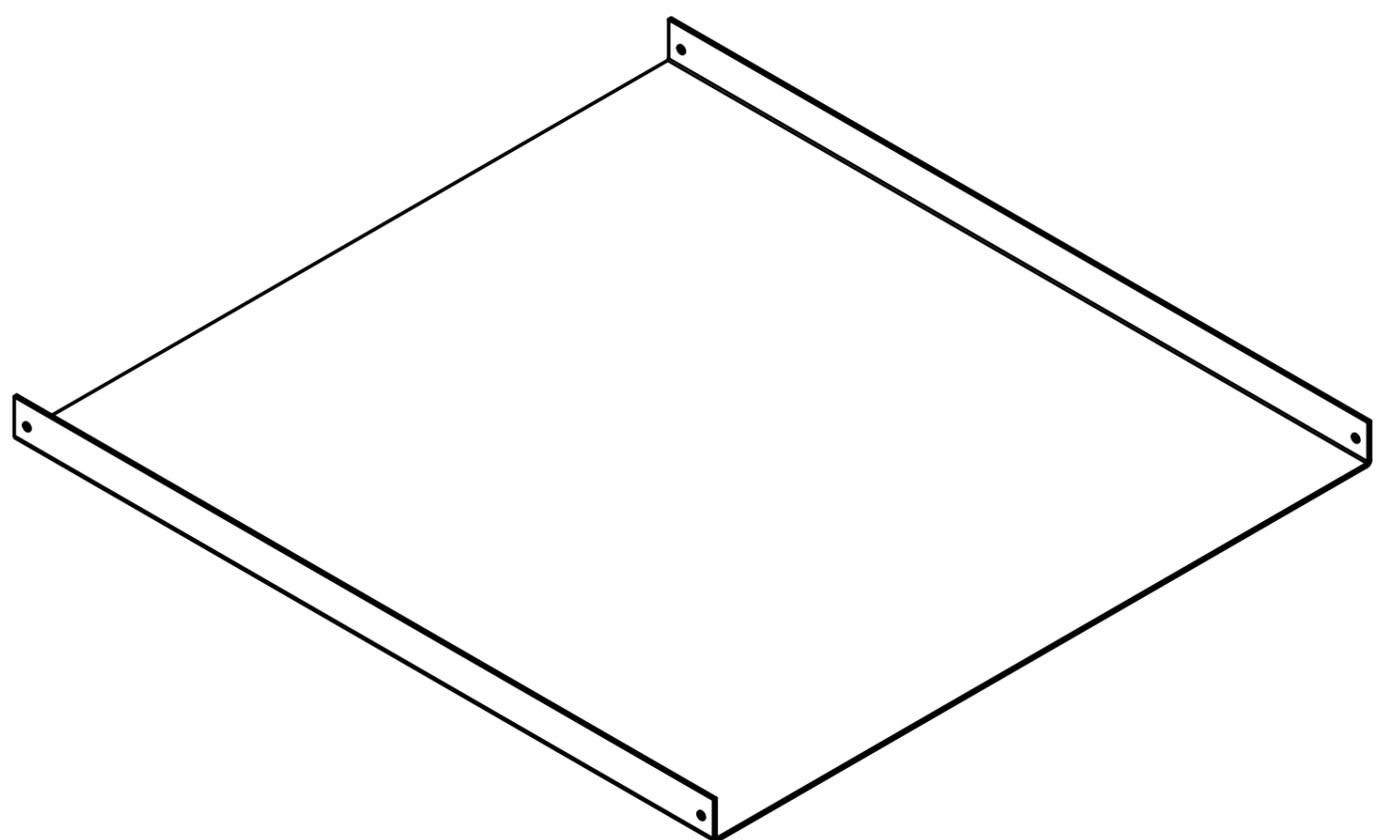
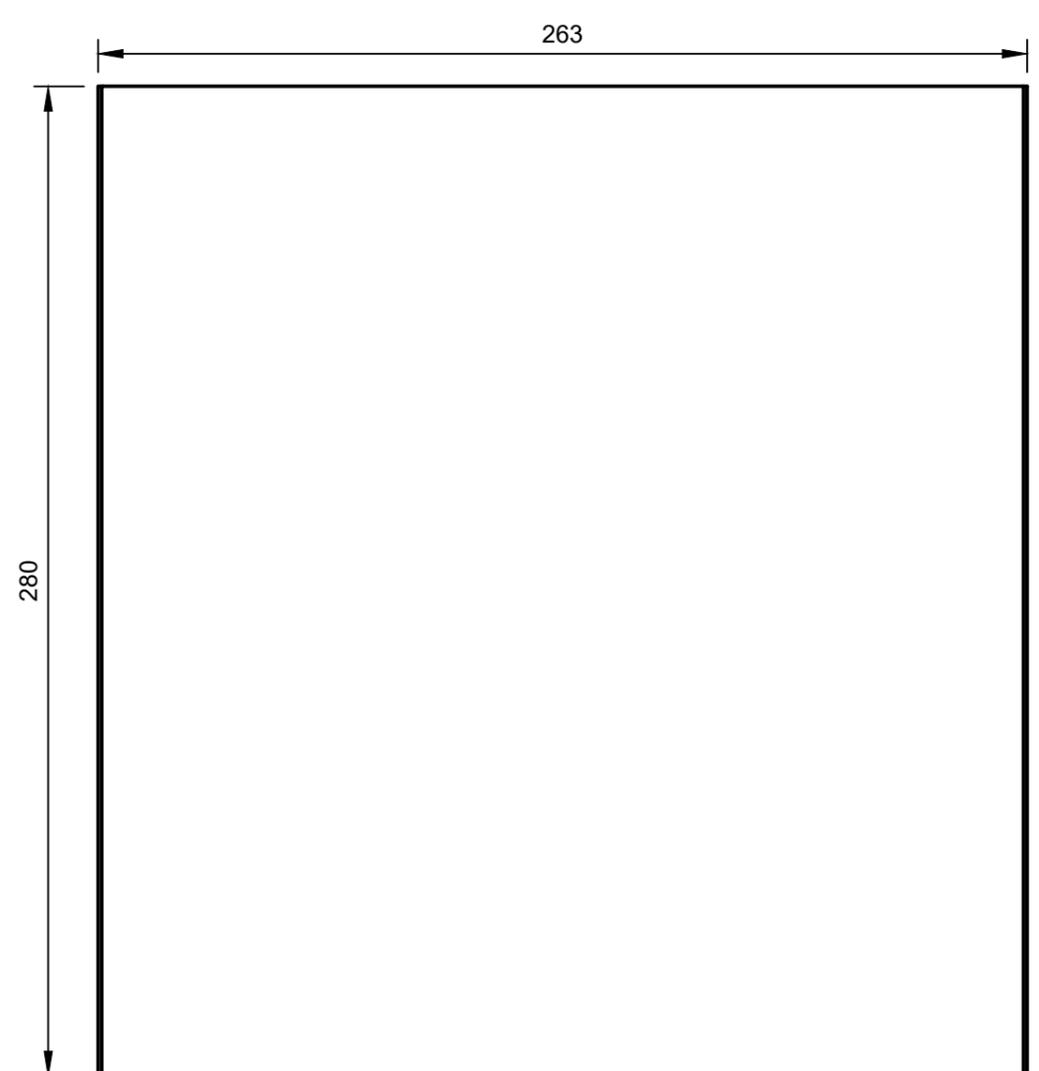
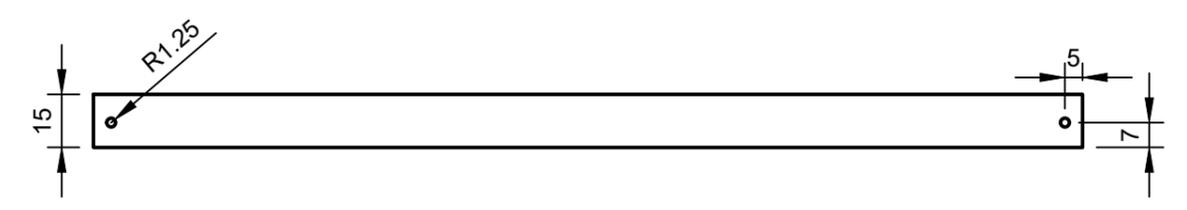
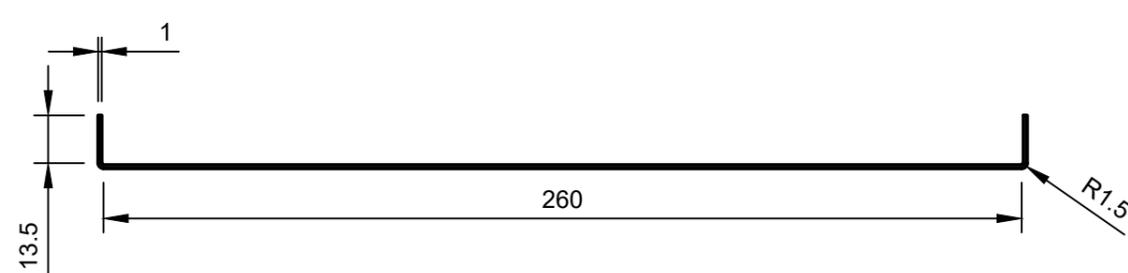
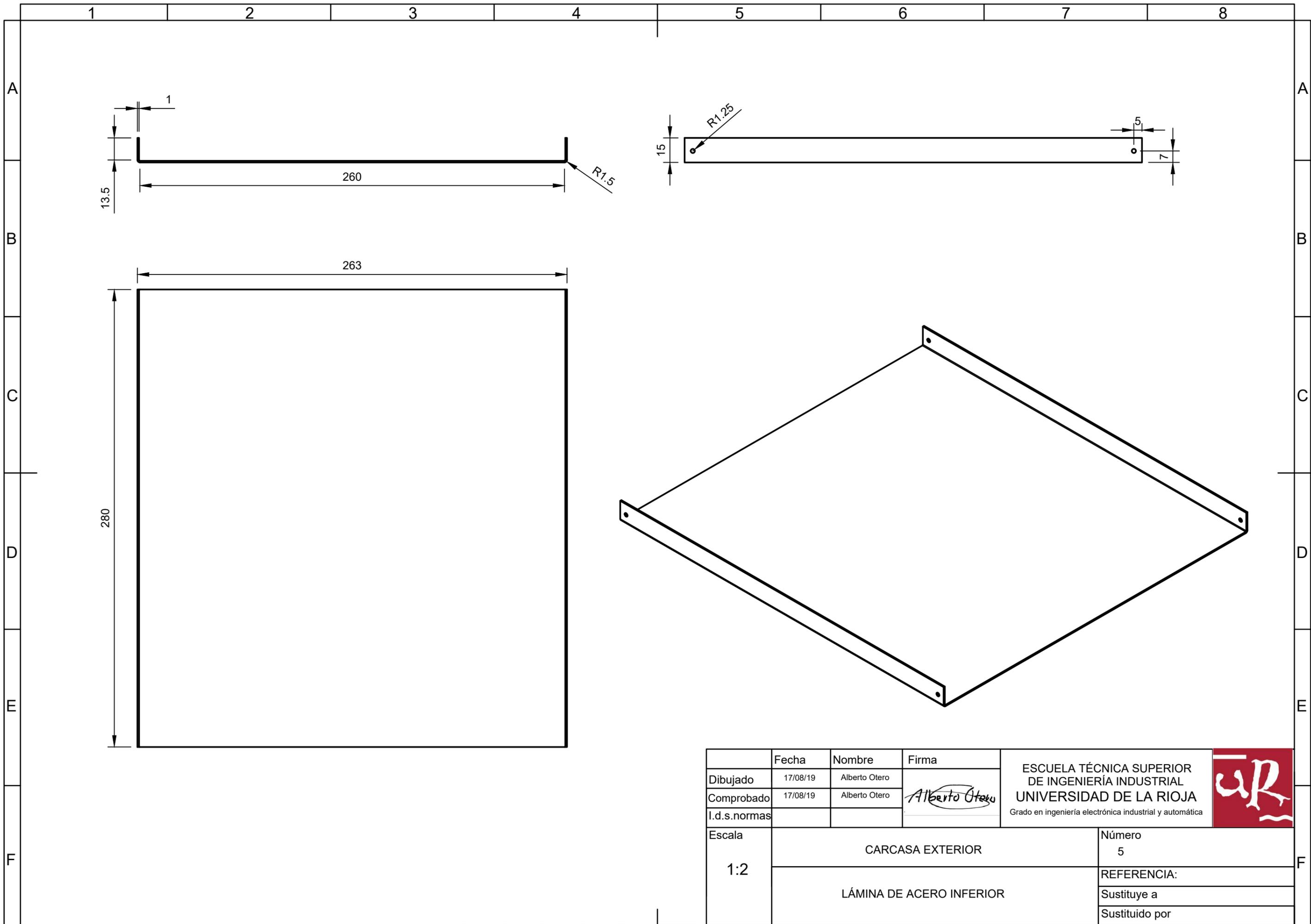
	Fecha	Nombre	Firma	ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL UNIVERSIDAD DE LA RIOJA <small>Grado en ingeniería electrónica industrial y automática</small>	
Dibujado	18/08/19	Alberto Otero			
Comprobado	18/08/19	Alberto Otero			
I.d.s.normas					
Escala	PANELES DE PLÁSTICO			Número	2
1:2	PANEL FRONTAL - VISTA EXTERIOR			REFERENCIA:	
				Sustituye a	
				Sustituido por	



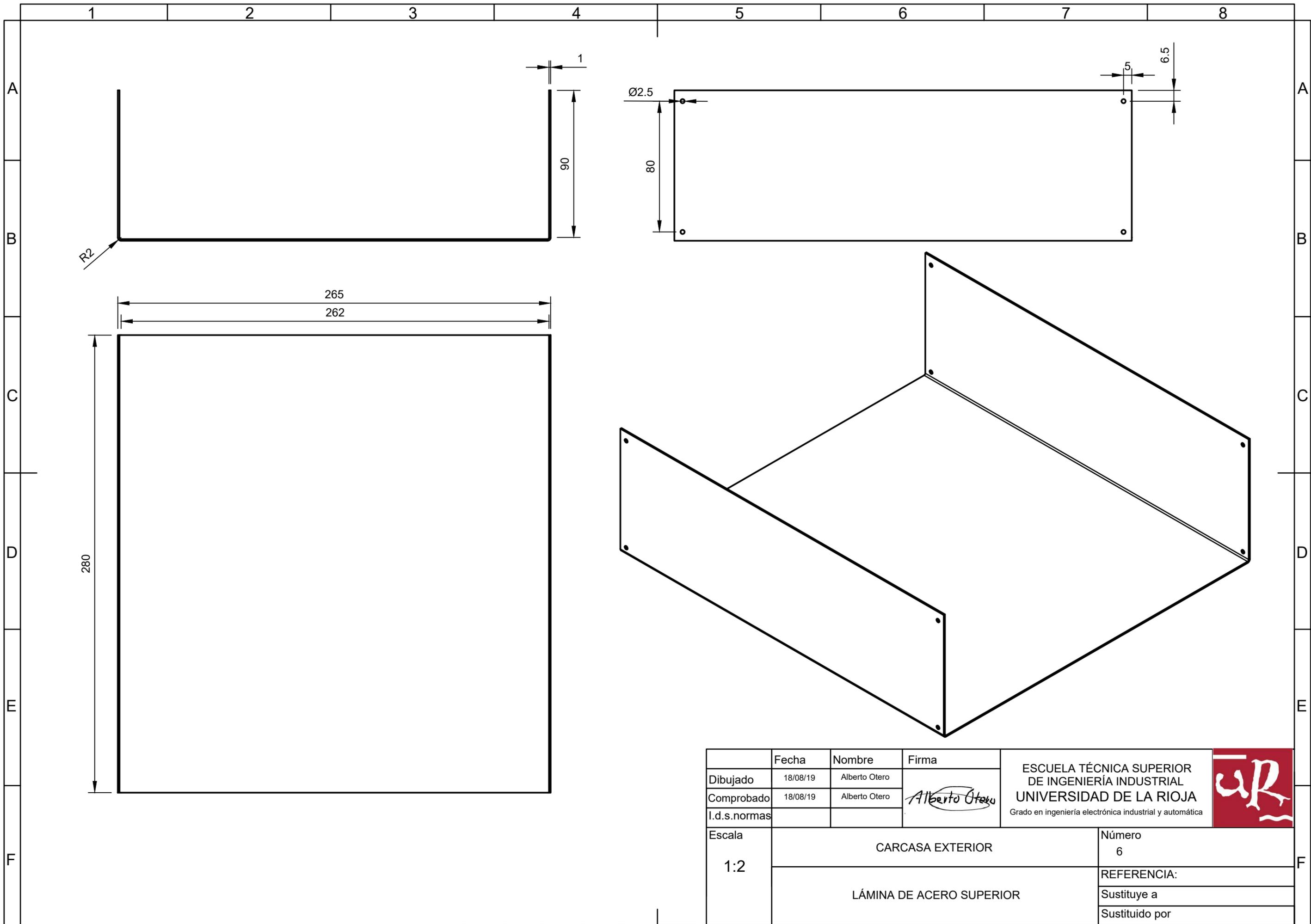
	Fecha	Nombre	Firma	ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL UNIVERSIDAD DE LA RIOJA <small>Grado en ingeniería electrónica industrial y automática</small>	
Dibujado	18/08/19	Alberto Otero			
Comprobado	18/08/19	Alberto Otero			
I.d.s.normas					
Escala	PANELES DE PLÁSTICO			Número	3
1:2	PANEL FRONTAL - VISTA INTERIOR			REFERENCIA:	
				Sustituye a	
				Sustituido por	



	Fecha	Nombre	Firma	ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL UNIVERSIDAD DE LA RIOJA <small>Grado en ingeniería electrónica industrial y automática</small>	
Dibujado	18/08/19	Alberto Otero			
Comprobado	18/08/19	Alberto Otero			
I.d.s.normas					
Escalas	1:2			Número 4	
	PANELES DE PLÁSTICO			REFERENCIA:	
	PANEL TRASERO			Sustituye a	
				Sustituido por	



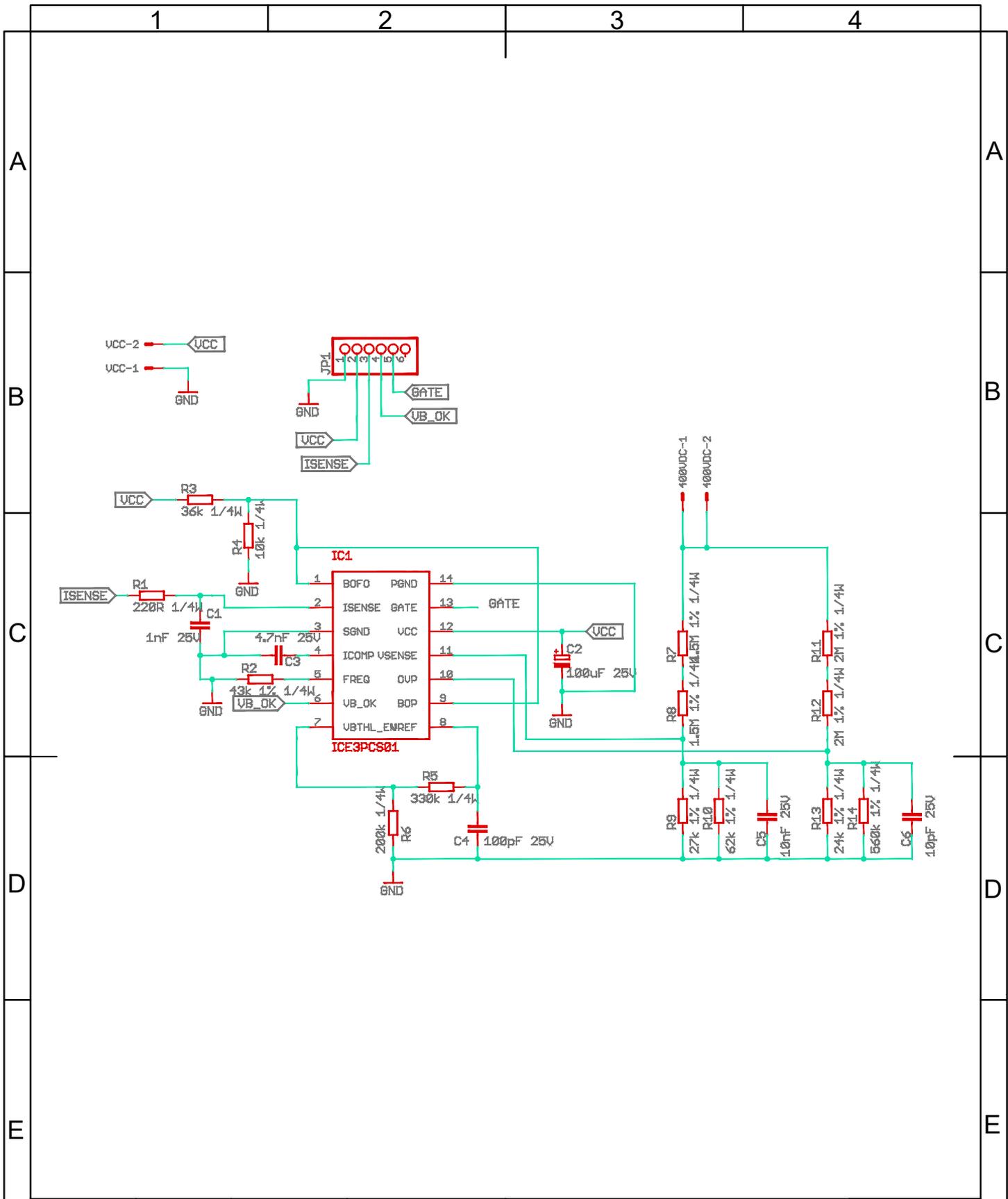
	Fecha	Nombre	Firma	ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL UNIVERSIDAD DE LA RIOJA <small>Grado en ingeniería electrónica industrial y automática</small>	
Dibujado	17/08/19	Alberto Otero			
Comprobado	17/08/19	Alberto Otero			
I.d.s.normas					
Escala	CARCASA EXTERIOR			Número	5
1:2	LÁMINA DE ACERO INFERIOR			REFERENCIA:	
				Sustituye a	
				Sustituido por	



	Fecha	Nombre	Firma	ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL UNIVERSIDAD DE LA RIOJA <small>Grado en ingeniería electrónica industrial y automática</small>	
Dibujado	18/08/19	Alberto Otero			
Comprobado	18/08/19	Alberto Otero			
I.d.s.normas					
Escala	CARCASA EXTERIOR			Número	6
1:2	LÁMINA DE ACERO SUPERIOR			REFERENCIA:	
				Sustituye a	
				Sustituido por	

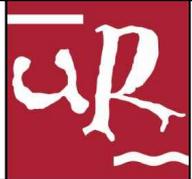
# PLANOS ELÉCTRICOS



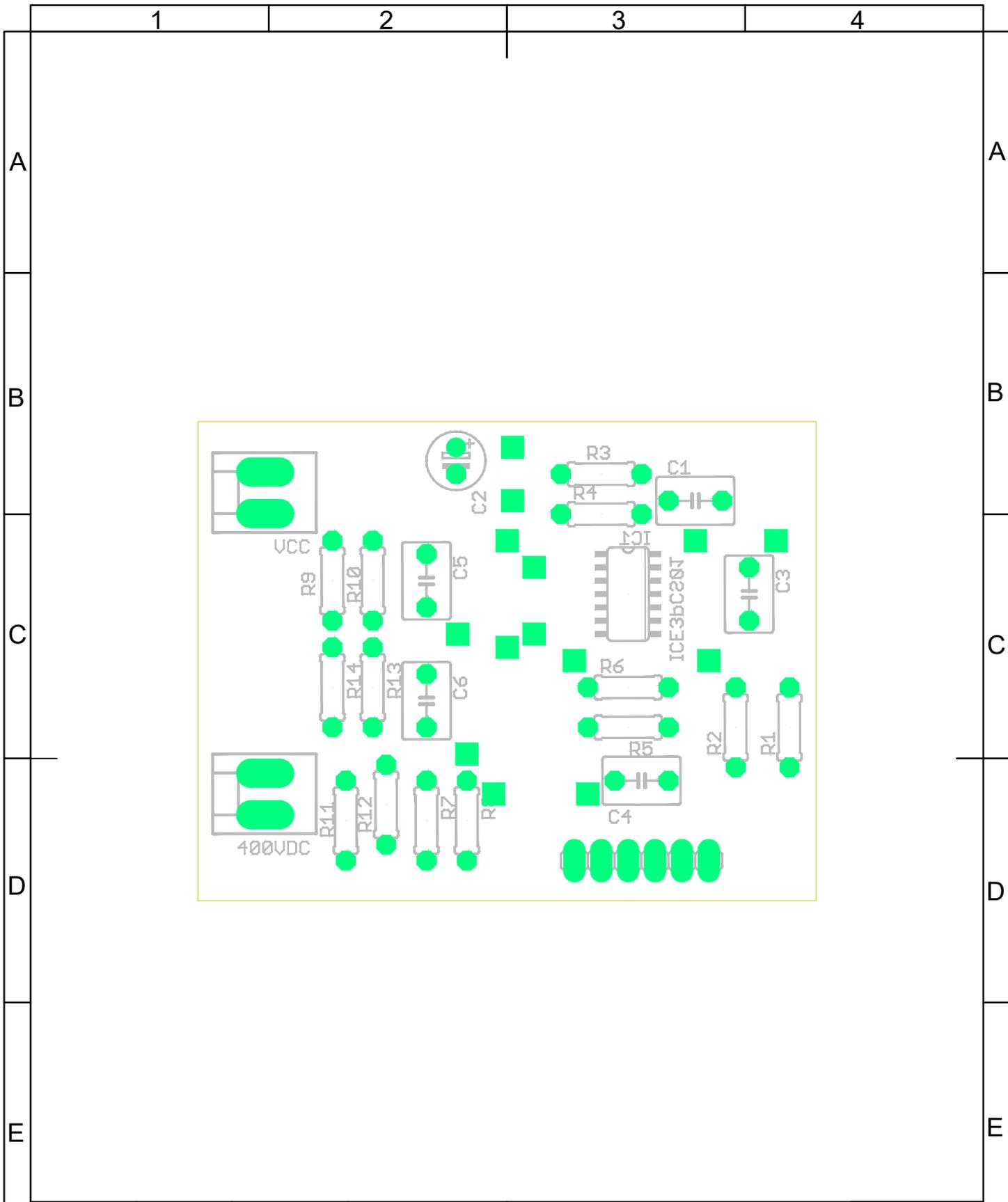


	Fecha	Nombre	Firma
Dibujado	21/08/19	Alberto Otero	
Comprobado	21/08/19	Alberto Otero	
I.d.s.normas			

ESCUELA TÉCNICA SUPERIOR  
 DE INGENIERÍA INDUSTRIAL  
 UNIVERSIDAD DE LA RIOJA  
 Grado en ingeniería electrónica industrial y automática



Escala Sin Escala	Controlador PFC		Número 1
	Esquema eléctrico		REFERENCIA:
			Sustituye a
		Sustituido por	

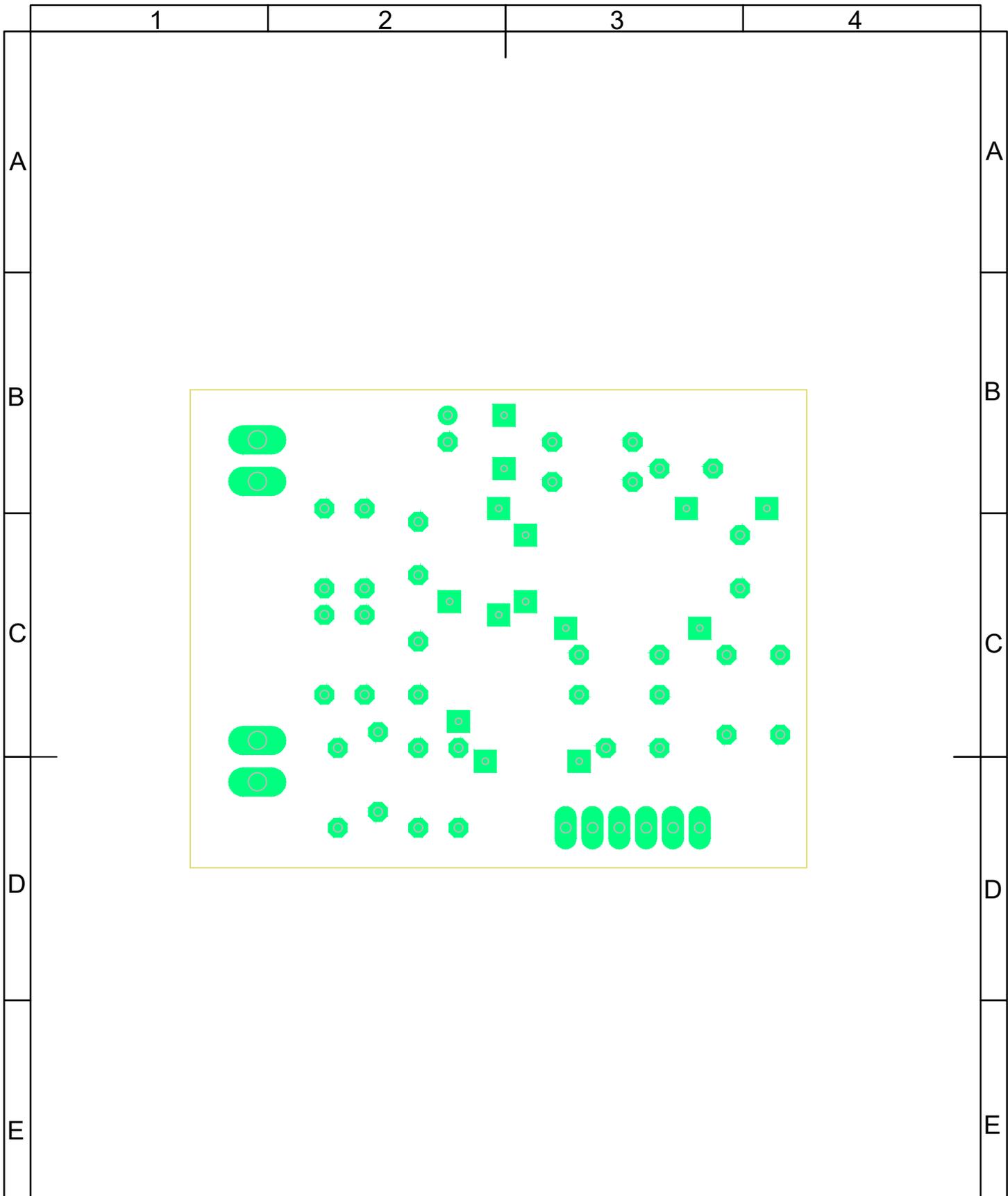


	Fecha	Nombre	Firma
Dibujado	21/08/19	Alberto Otero	
Comprobado	21/08/19	Alberto Otero	
I.d.s.normas			

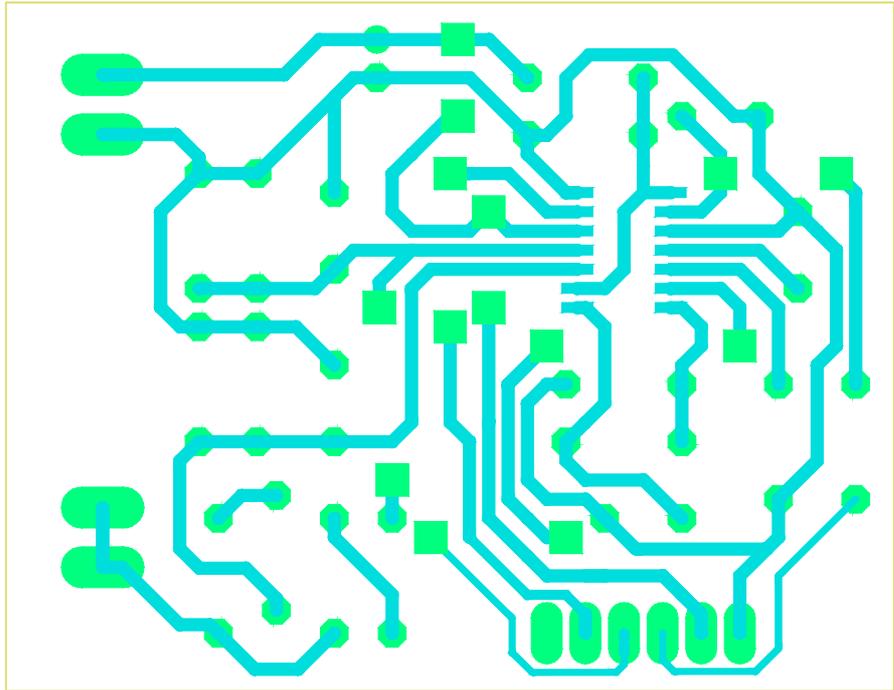
ESCUELA TÉCNICA SUPERIOR  
 DE INGENIERÍA INDUSTRIAL  
 UNIVERSIDAD DE LA RIOJA  
 Grado en ingeniería electrónica industrial y automática



Escala 2:1	Controlador PFC	Número 2
	Layout componentes	REFERENCIA:
		Sustituye a
		Sustituido por

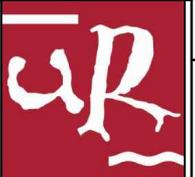


	Fecha	Nombre	Firma	ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL <b>UNIVERSIDAD DE LA RIOJA</b> <small>Grado en ingeniería electrónica industrial y automática</small>	
Dibujado	21/08/19	Alberto Otero			
Comprobado	21/08/19	Alberto Otero			
I.d.s.normas					
Escala 2:1	Controlador PFC			Número 3	
	Layout Taladros			REFERENCIA:	
				Sustituye a	
			Sustituido por		

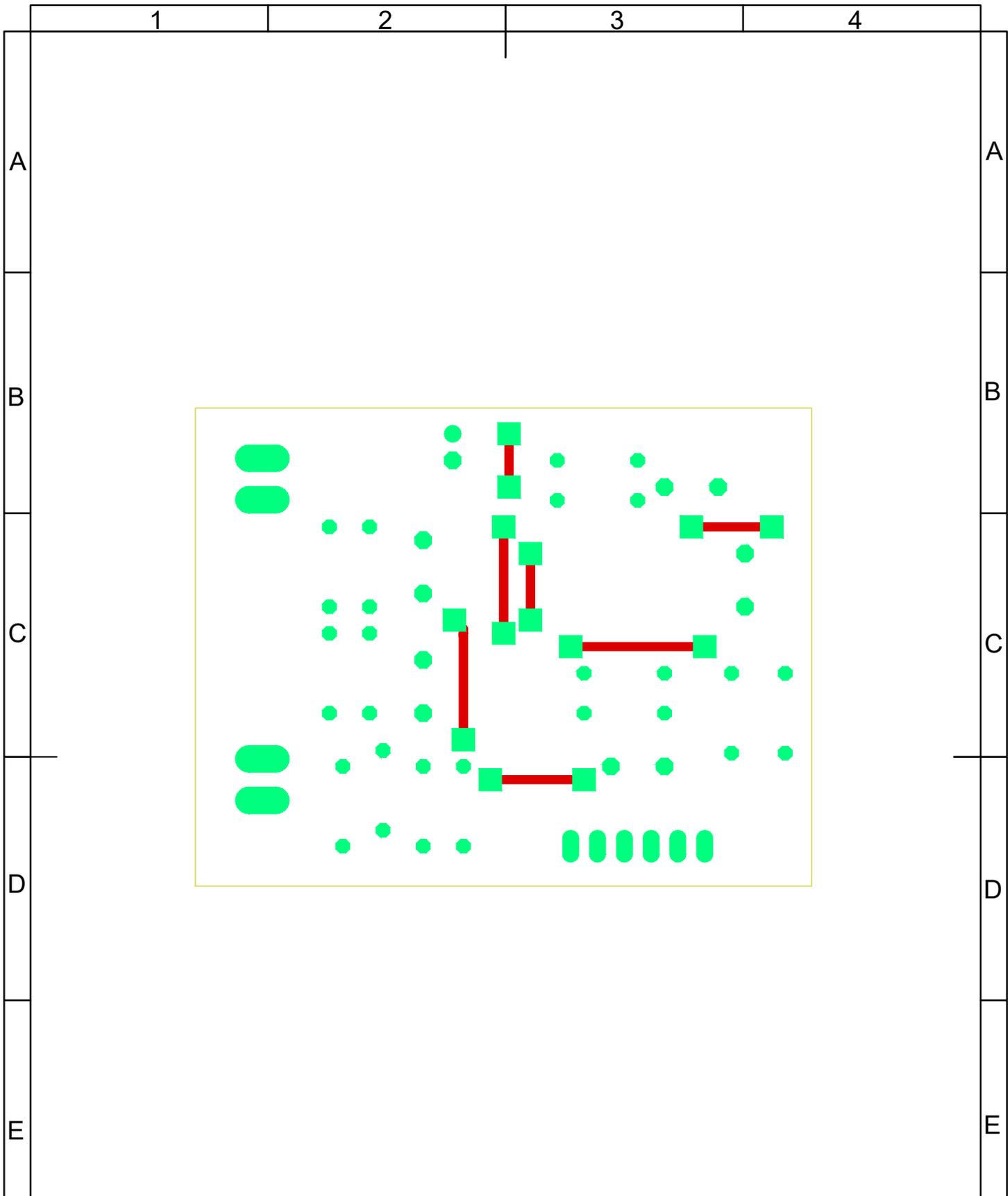


	Fecha	Nombre	Firma
Dibujado	21/08/19	Alberto Otero	
Comprobado	21/08/19	Alberto Otero	
I.d.s.normas			

ESCUELA TÉCNICA SUPERIOR  
 DE INGENIERÍA INDUSTRIAL  
 UNIVERSIDAD DE LA RIOJA  
 Grado en ingeniería electrónica industrial y automática



Escala 2:1	Controlador PFC	Número 4
	Pistas - Capa inferior	REFERENCIA:
		Sustituye a
		Sustituido por

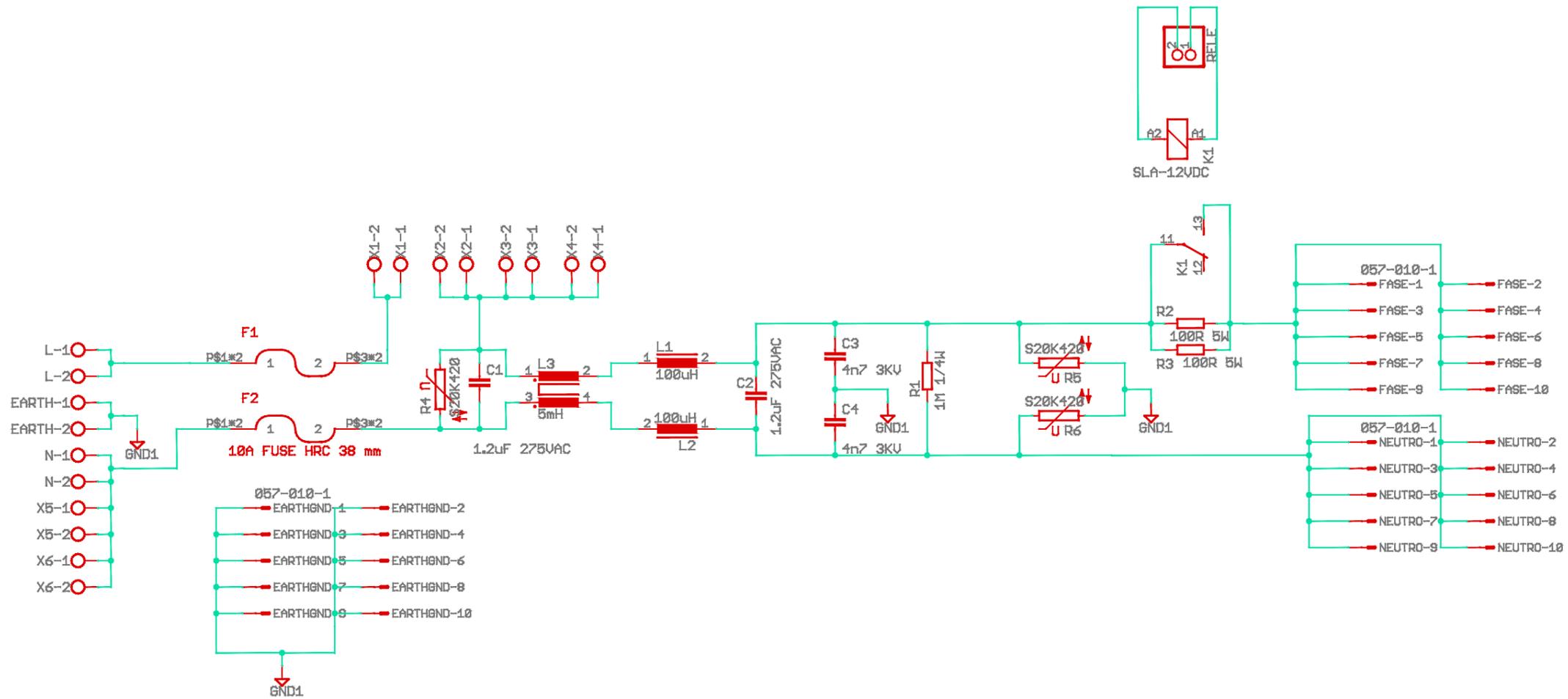


	Fecha	Nombre	Firma
Dibujado	21/08/19	Alberto Otero	
Comprobado	21/08/19	Alberto Otero	
I.d.s.normas			

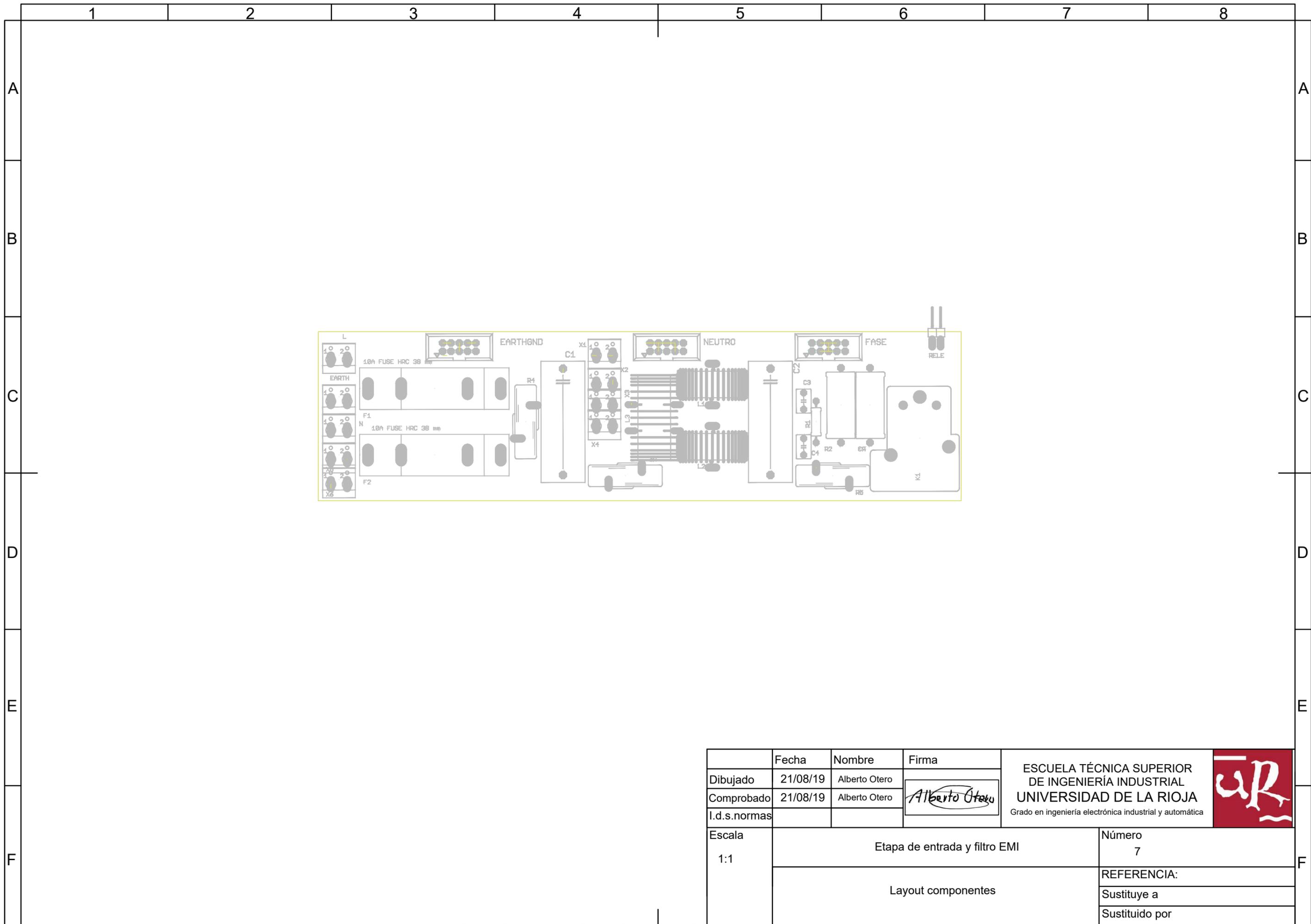
ESCUELA TÉCNICA SUPERIOR  
 DE INGENIERÍA INDUSTRIAL  
 UNIVERSIDAD DE LA RIOJA  
 Grado en ingeniería electrónica industrial y automática



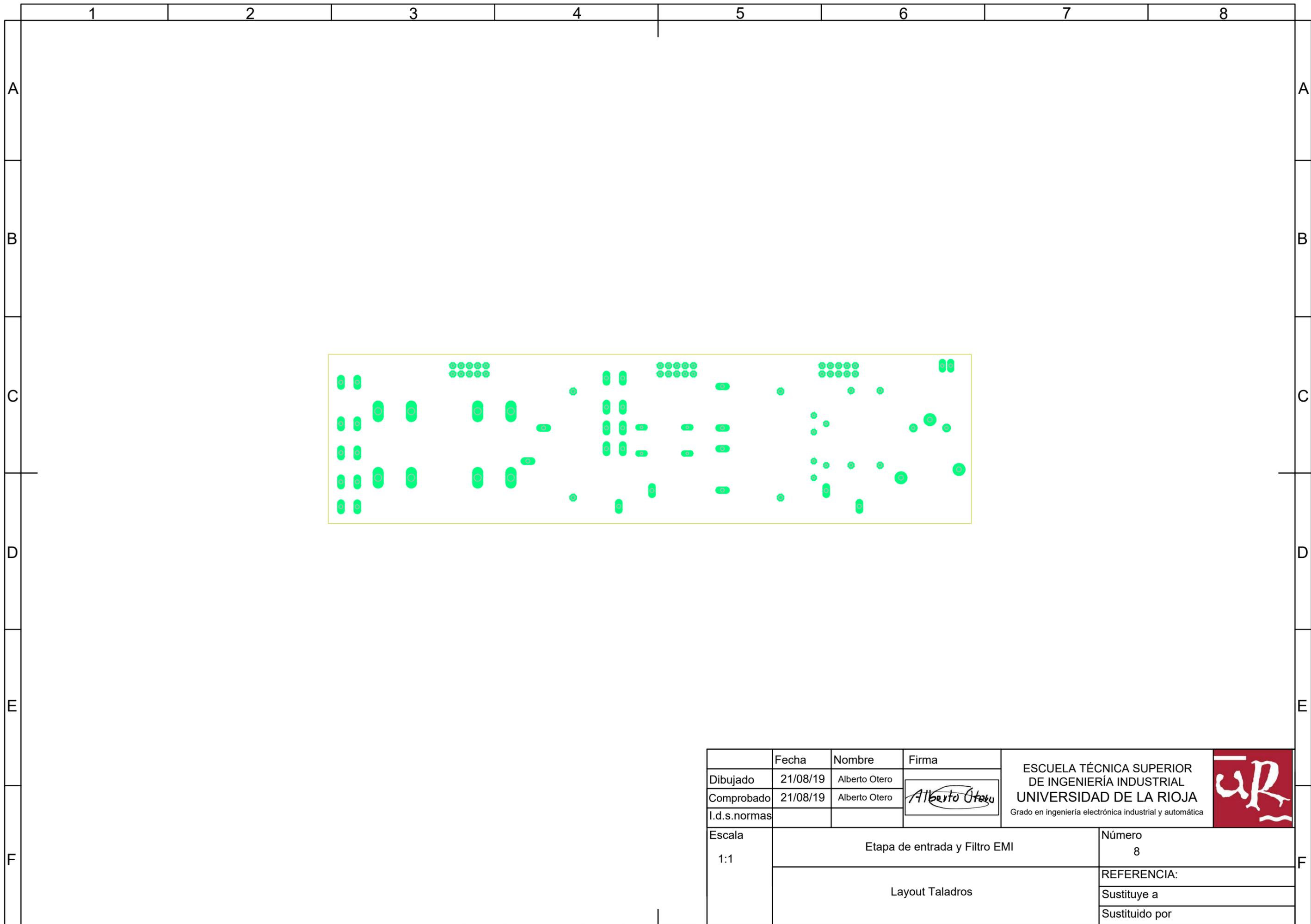
Escala 2:1	Controlador PFC	Número 5
	Pistas - Capa superior	REFERENCIA:
		Sustituye a
		Sustituido por



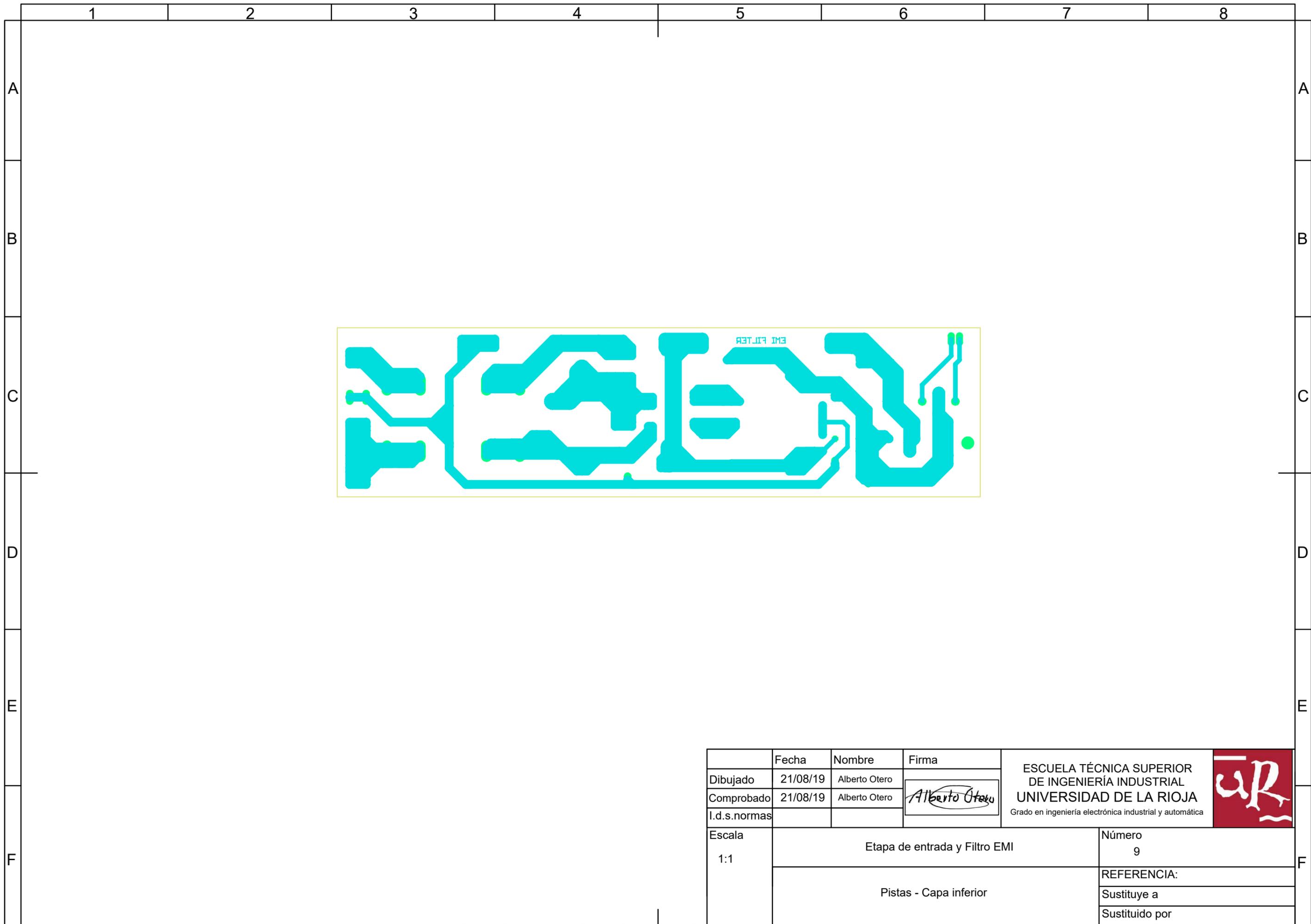
	Fecha	Nombre	Firma	ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL UNIVERSIDAD DE LA RIOJA Grado en ingeniería electrónica industrial y automática	
Dibujado	21/08/19	Alberto Otero			
Comprobado	21/08/19	Alberto Otero			
I.d.s.normas					
Escala	Etapa de entrada y filtro EMI			Número	6
Sin Escala	Esquema eléctrico			REFERENCIA:	
				Sustituye a	
				Sustituido por	

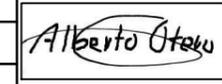


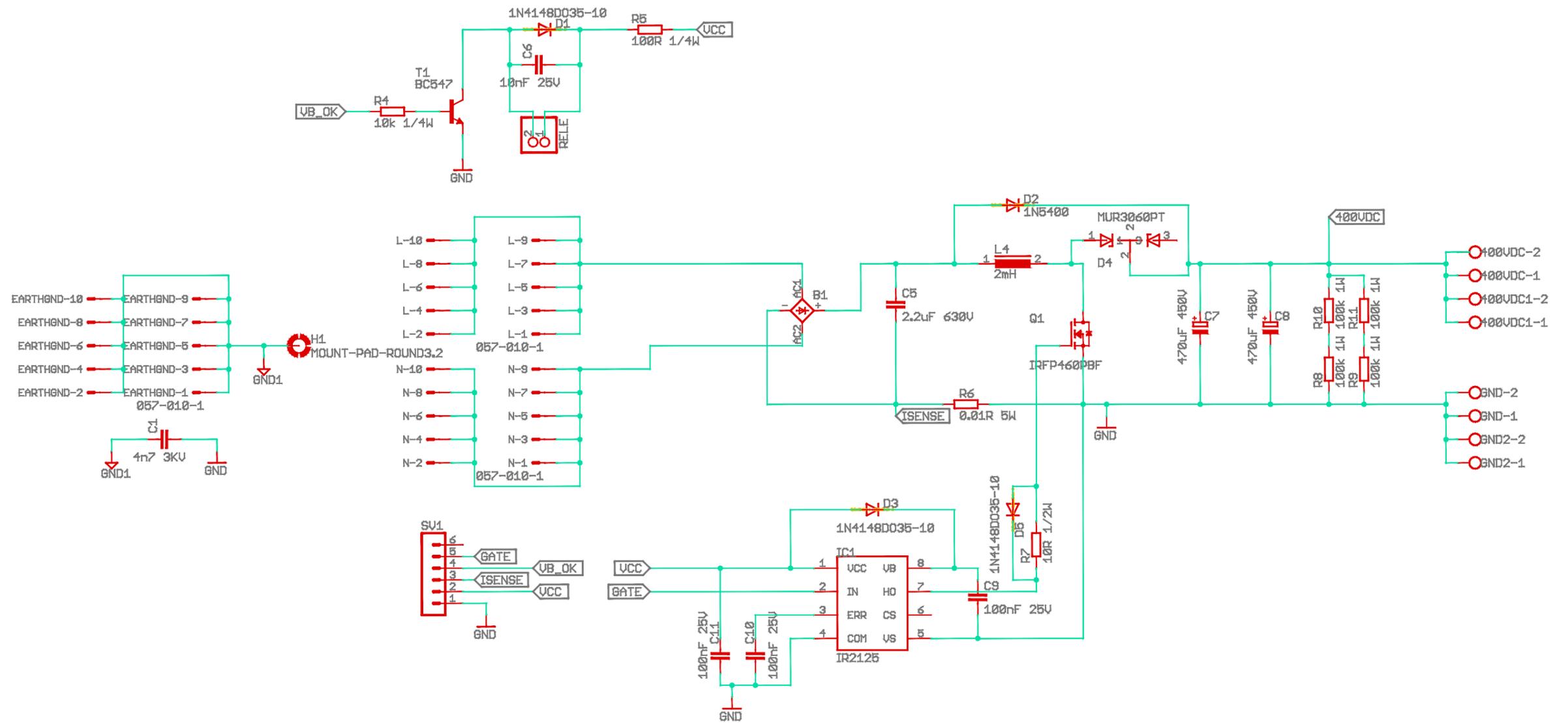
	Fecha	Nombre	Firma	ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL UNIVERSIDAD DE LA RIOJA <small>Grado en ingeniería electrónica industrial y automática</small> 
Dibujado	21/08/19	Alberto Otero		
Comprobado	21/08/19	Alberto Otero		
I.d.s.normas				
Escala	Etapa de entrada y filtro EMI			Número 7
1:1	Layout componentes			REFERENCIA:
				Sustituye a
				Sustituido por



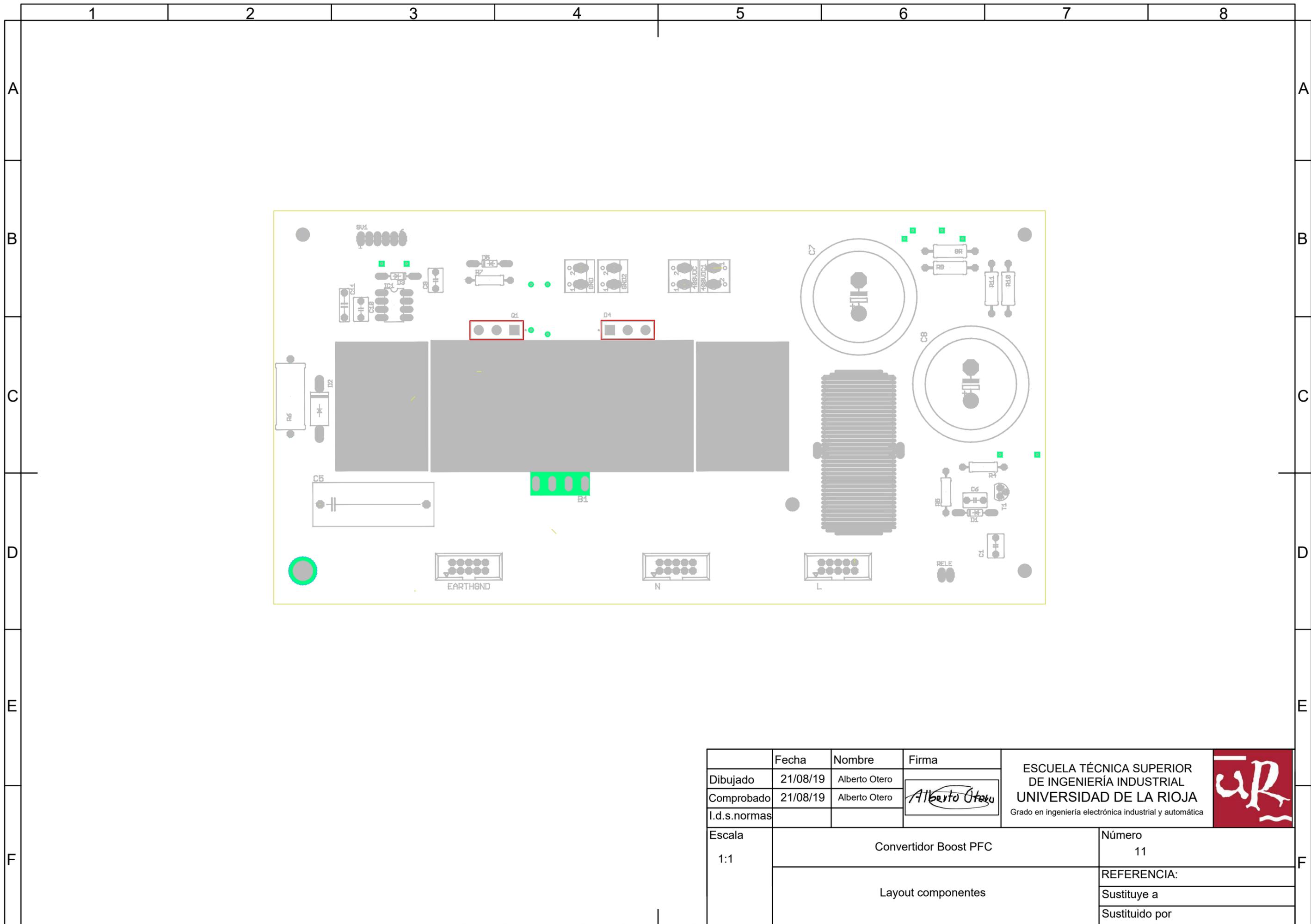
	Fecha	Nombre	Firma	ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL UNIVERSIDAD DE LA RIOJA <small>Grado en ingeniería electrónica industrial y automática</small> 
Dibujado	21/08/19	Alberto Otero		
Comprobado	21/08/19	Alberto Otero		
I.d.s.normas				
Escala	Etapa de entrada y Filtro EMI			Número 8
1:1	Layout Taladros			REFERENCIA:
				Sustituye a
				Sustituido por



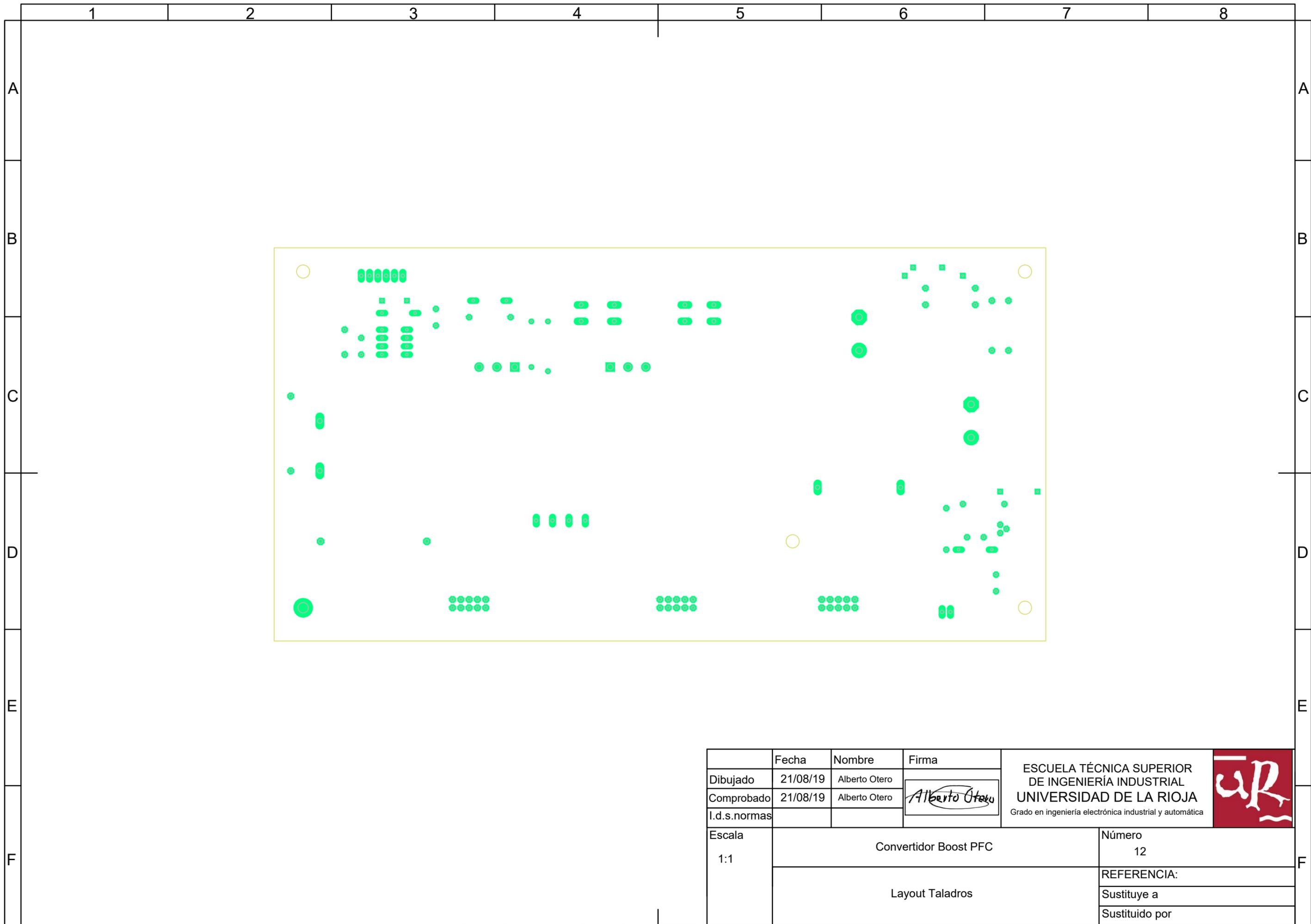
	Fecha	Nombre	Firma	ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL UNIVERSIDAD DE LA RIOJA <small>Grado en ingeniería electrónica industrial y automática</small> 
Dibujado	21/08/19	Alberto Otero		
Comprobado	21/08/19	Alberto Otero		
I.d.s.normas				
Escala	Etapa de entrada y Filtro EMI		Número	
1:1	Pistas - Capa inferior		9	
			REFERENCIA:	
			Sustituye a	
			Sustituido por	



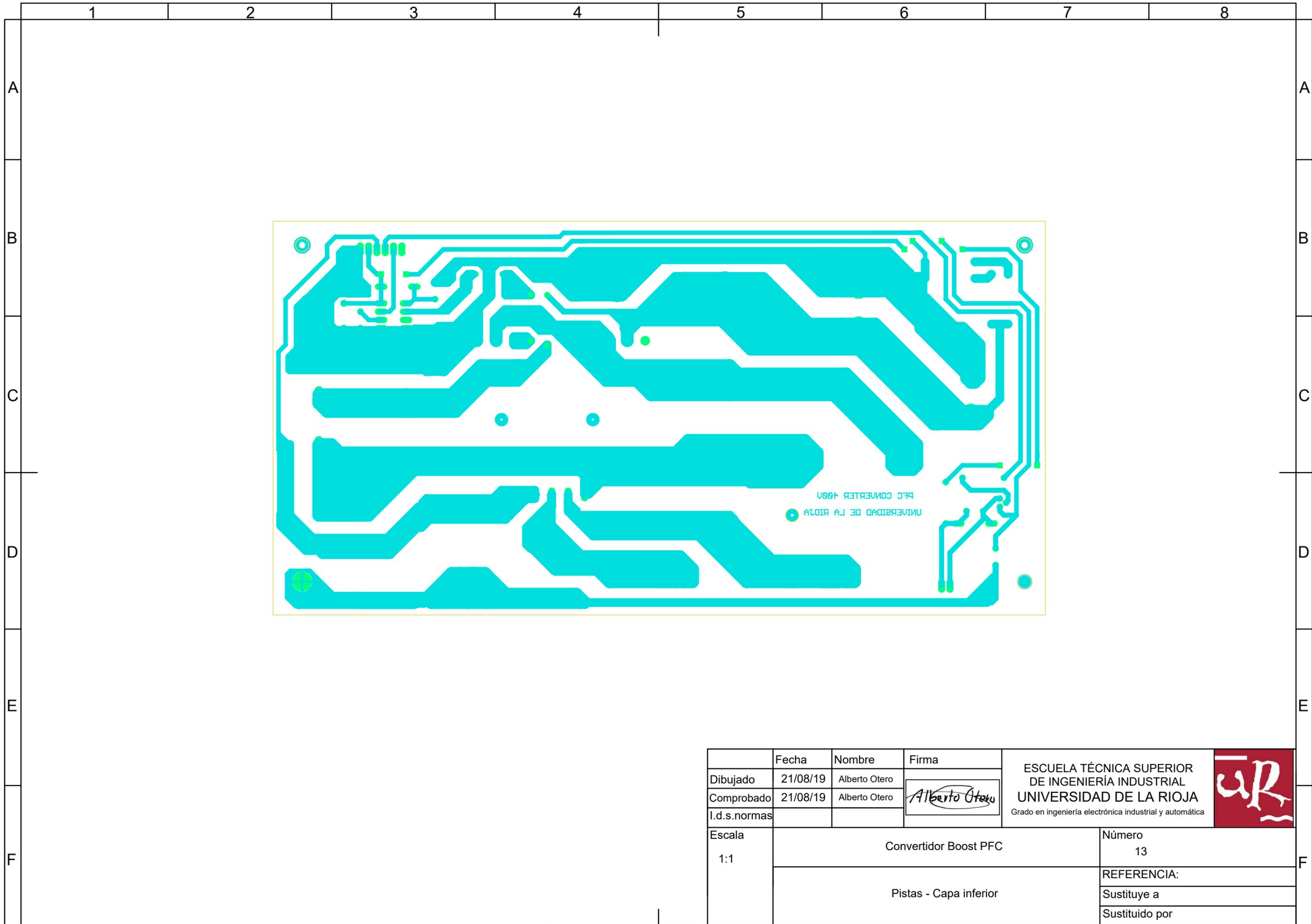
	Fecha	Nombre	Firma	ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL UNIVERSIDAD DE LA RIOJA Grado en ingeniería electrónica industrial y automática	
Dibujado	21/08/19	Alberto Otero			
Comprobado	21/08/19	Alberto Otero			
I.d.s.normas					
Escala	Convertidor Boost PFC			Número	10
Sin Escala	Esquema eléctrico			REFERENCIA:	
				Sustituye a	
				Sustituido por	



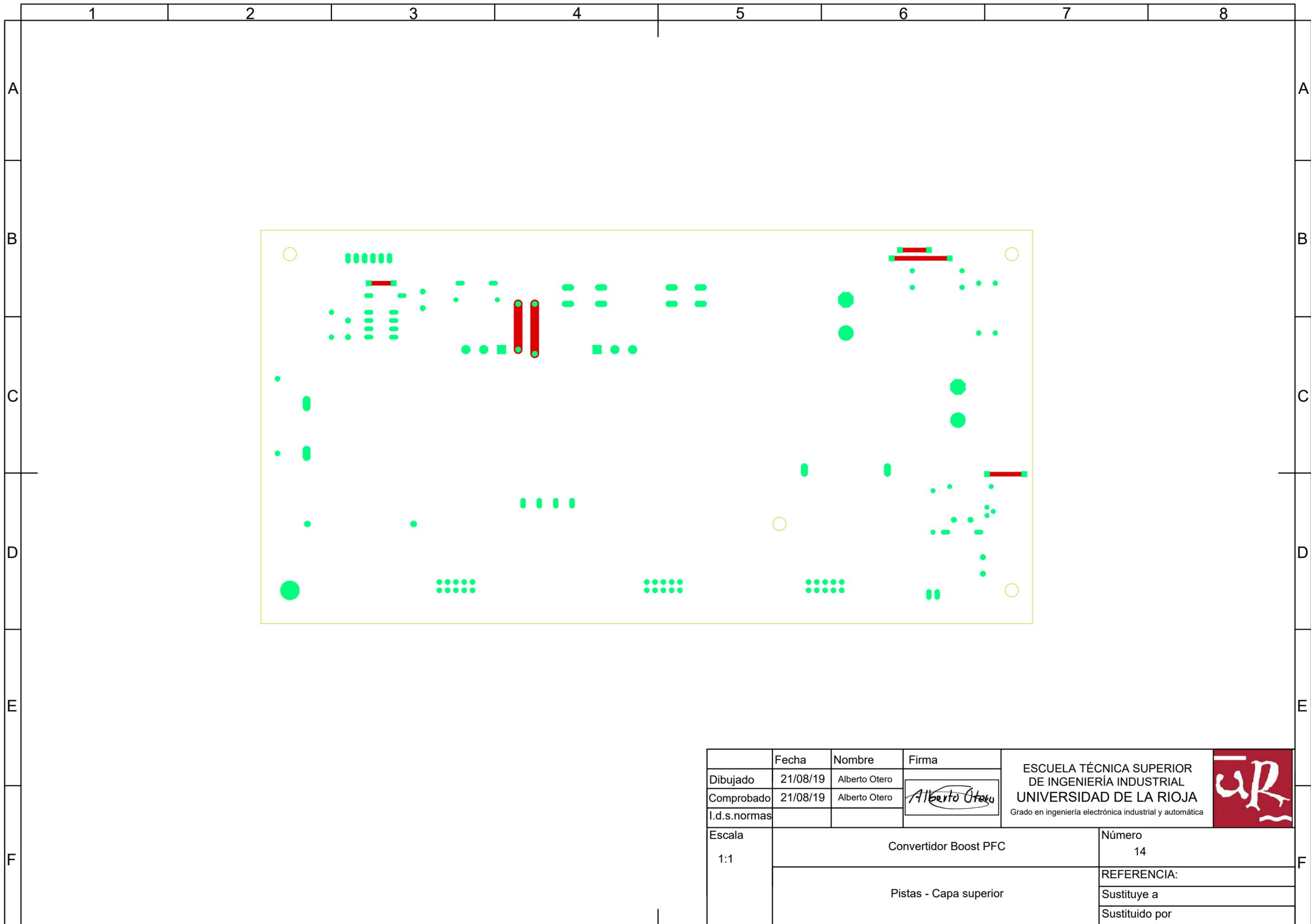
	Fecha	Nombre	Firma	ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL UNIVERSIDAD DE LA RIOJA <small>Grado en ingeniería electrónica industrial y automática</small> 
Dibujado	21/08/19	Alberto Otero		
Comprobado	21/08/19	Alberto Otero		
I.d.s.normas				
Escala	Convertidor Boost PFC			Número 11
1:1	Layout componentes			REFERENCIA:
				Sustituye a
				Sustituido por



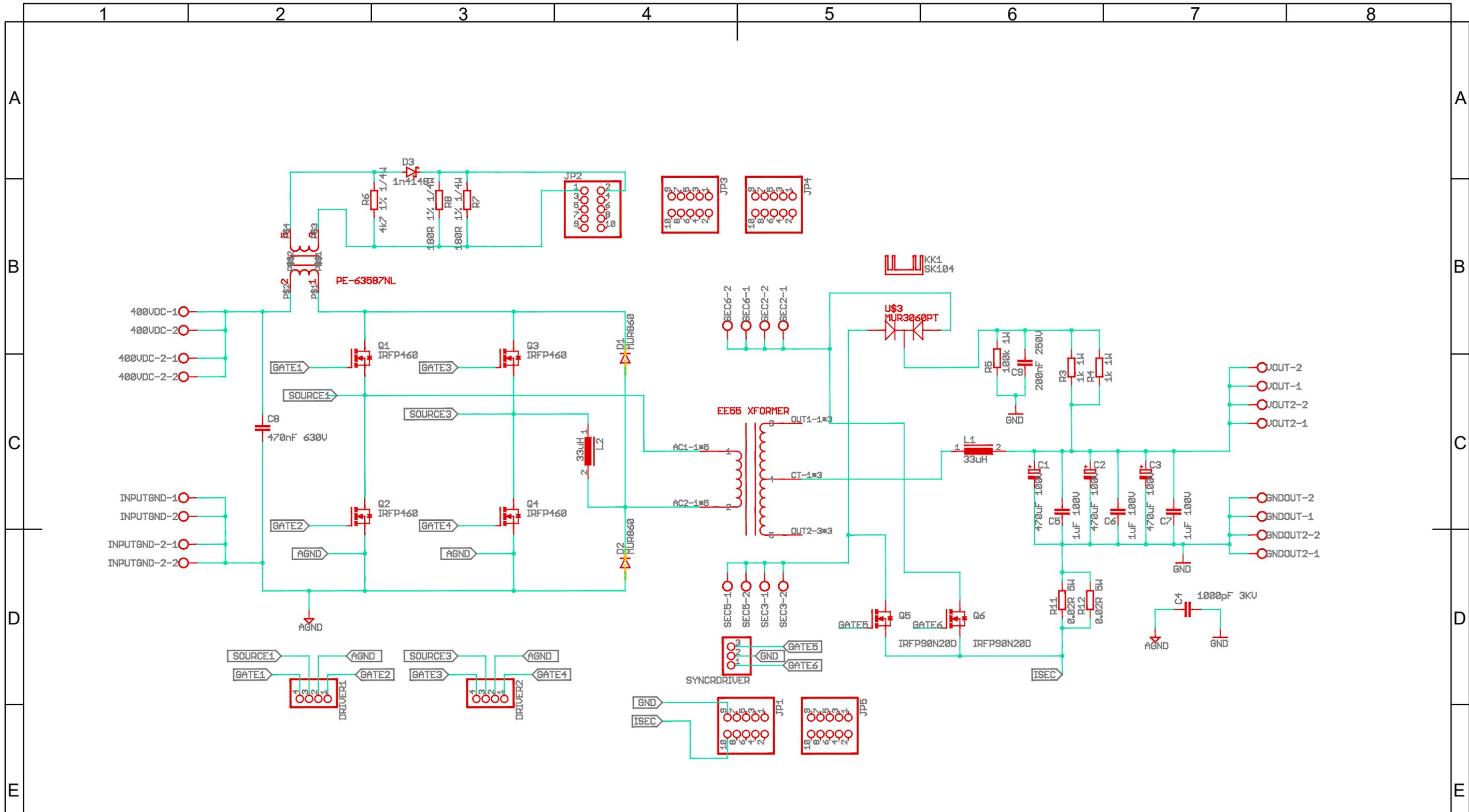
	Fecha	Nombre	Firma	ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL UNIVERSIDAD DE LA RIOJA <small>Grado en ingeniería electrónica industrial y automática</small> 
Dibujado	21/08/19	Alberto Otero		
Comprobado	21/08/19	Alberto Otero		
I.d.s.normas				
Escala	Convertidor Boost PFC			Número 12
1:1	Layout Taladros			REFERENCIA:
				Sustituye a
				Sustituido por

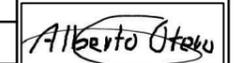


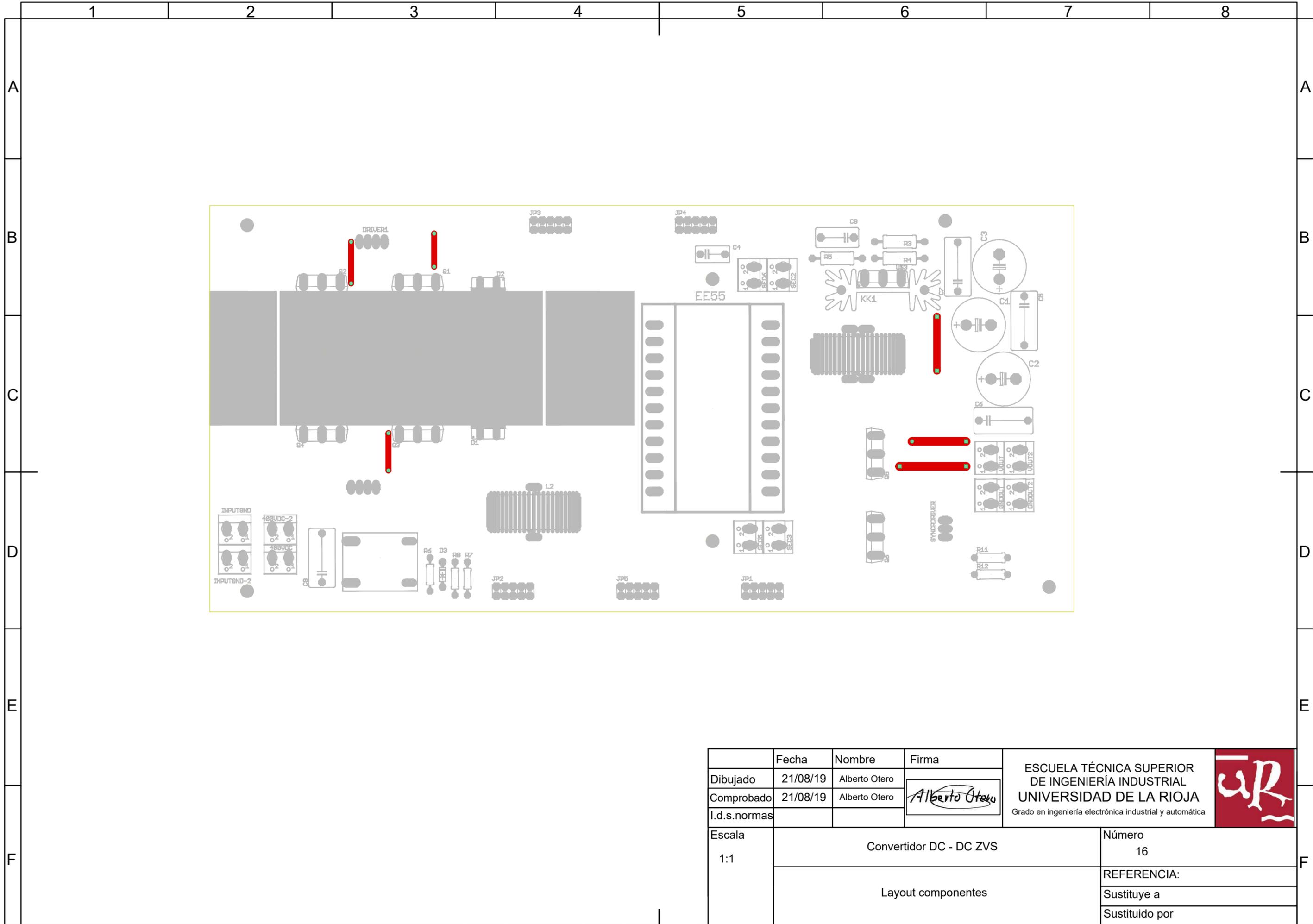
	Fecha	Nombre	Firma	ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL UNIVERSIDAD DE LA RIOJA <small>Grado en ingeniería electrónica industrial y automática</small> 
Dibujado	21/08/19	Alberto Otero		
Comprobado	21/08/19	Alberto Otero		
I.d.s.normas				
Escala	Convertidor Boost PFC		Número	
1:1	Pistas - Capa inferior		13	
			REFERENCIA:	
			Sustituye a	
			Sustituido por	



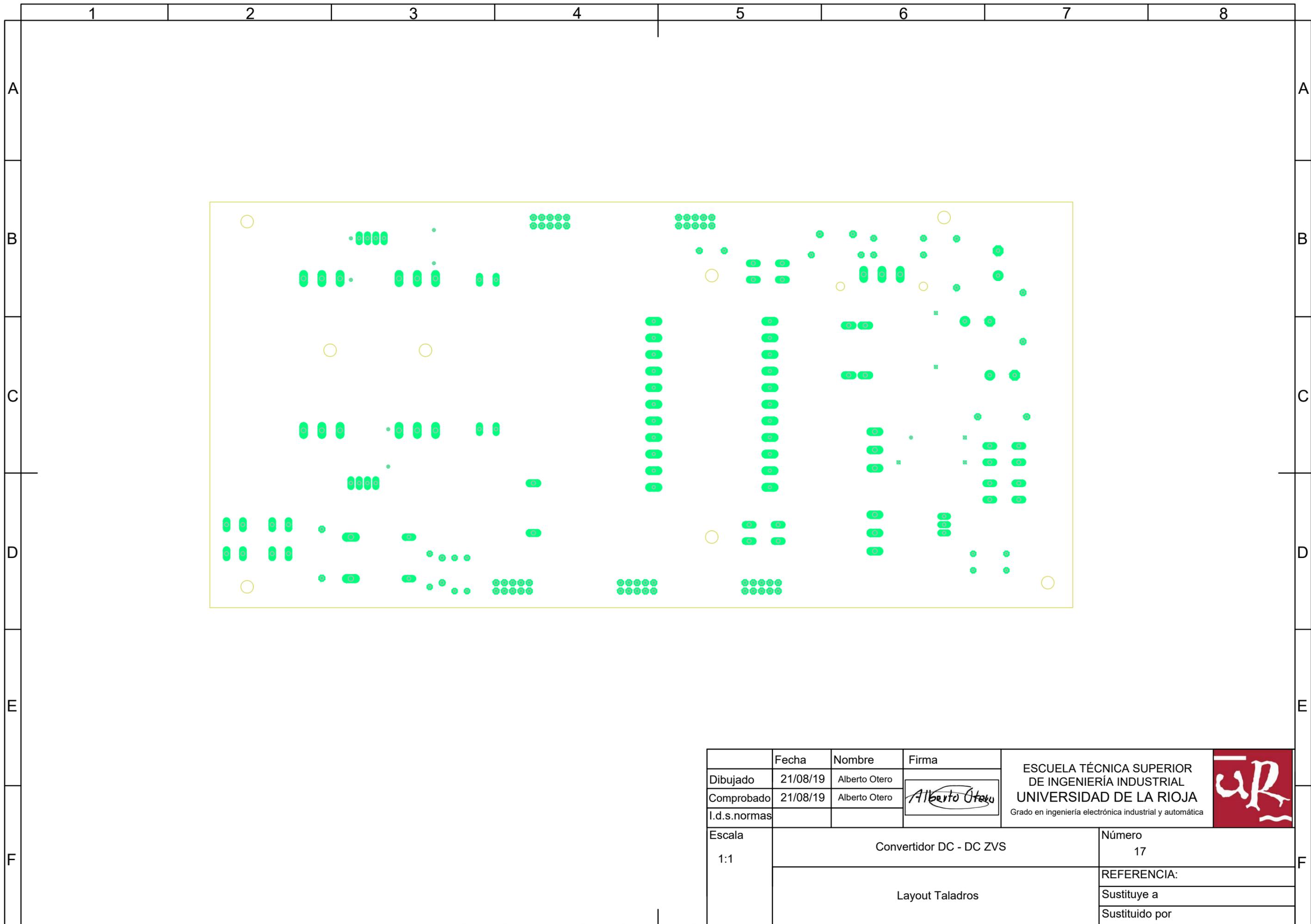
	Fecha	Nombre	Firma	ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL UNIVERSIDAD DE LA RIOJA <small>Grado en ingeniería electrónica industrial y automática</small> 
Dibujado	21/08/19	Alberto Otero		
Comprobado	21/08/19	Alberto Otero		
I.d.s.normas				
Escala	Convertidor Boost PFC			Número 14
1:1	Pistas - Capa superior			REFERENCIA:
				Sustituye a
				Sustituido por



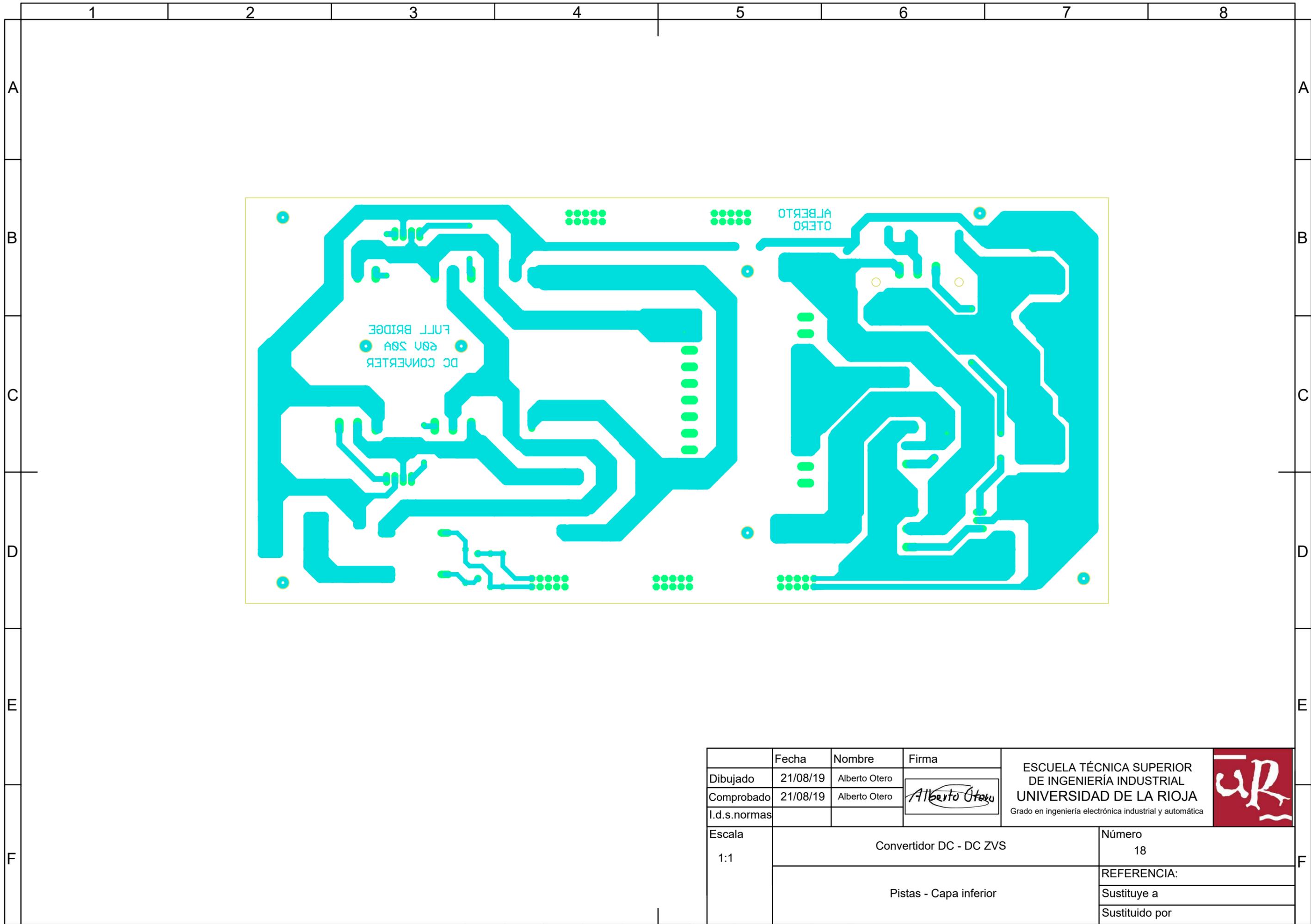
	Fecha	Nombre	Firma	ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL UNIVERSIDAD DE LA RIOJA Grado en ingeniería electrónica industrial y automática	
Dibujado	21/08/19	Alberto Otero			
Comprobado	21/08/19	Alberto Otero			
I.d.s.normas					
Escala	Convertidor DC - DC ZVS			Número	15
Sin Escala	Esquema eléctrico			REFERENCIA:	
				Sustituye a	
				Sustituido por	



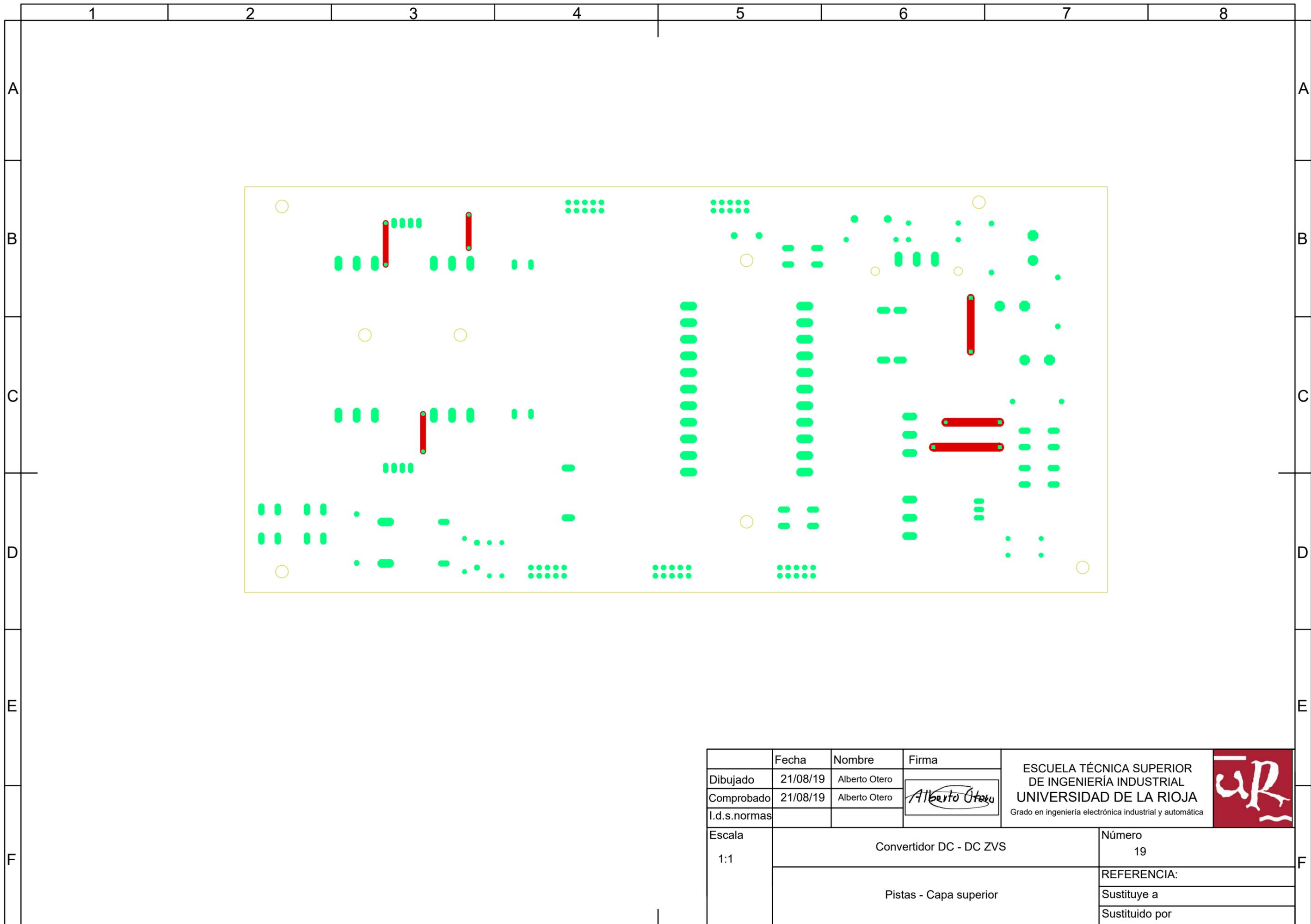
	Fecha	Nombre	Firma	ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL UNIVERSIDAD DE LA RIOJA <small>Grado en ingeniería electrónica industrial y automática</small> 
Dibujado	21/08/19	Alberto Otero		
Comprobado	21/08/19	Alberto Otero		
I.d.s.normas				
Escala	Convertidor DC - DC ZVS			Número 16
1:1	Layout componentes			REFERENCIA:
				Sustituye a
				Sustituido por



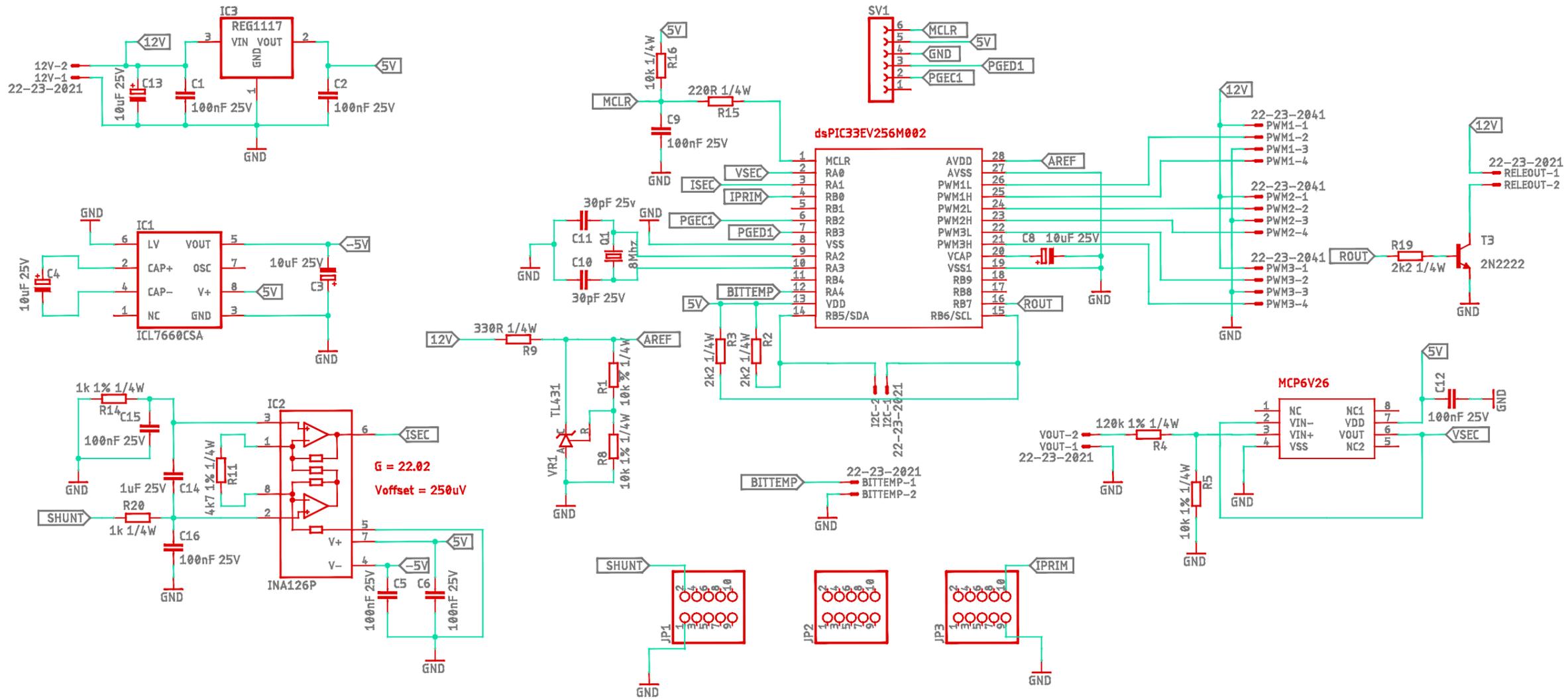
	Fecha	Nombre	Firma	ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL UNIVERSIDAD DE LA RIOJA <small>Grado en ingeniería electrónica industrial y automática</small> 
Dibujado	21/08/19	Alberto Otero		
Comprobado	21/08/19	Alberto Otero		
I.d.s.normas				
Escala	Convertidor DC - DC ZVS			Número 17
1:1	Layout Taladros			REFERENCIA:
				Sustituye a
				Sustituido por

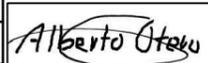


	Fecha	Nombre	Firma	ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL UNIVERSIDAD DE LA RIOJA <small>Grado en ingeniería electrónica industrial y automática</small> 
Dibujado	21/08/19	Alberto Otero		
Comprobado	21/08/19	Alberto Otero	<i>Alberto Otero</i>	
I.d.s.normas				
Escala	Convertidor DC - DC ZVS			Número 18
1:1	Pistas - Capa inferior			REFERENCIA:
				Sustituye a
				Sustituido por



	Fecha	Nombre	Firma	ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL UNIVERSIDAD DE LA RIOJA <small>Grado en ingeniería electrónica industrial y automática</small> 
Dibujado	21/08/19	Alberto Otero		
Comprobado	21/08/19	Alberto Otero		
I.d.s.normas				
Escala	Convertidor DC - DC ZVS			Número 19
1:1	Pistas - Capa superior			REFERENCIA:
				Sustituye a
				Sustituido por

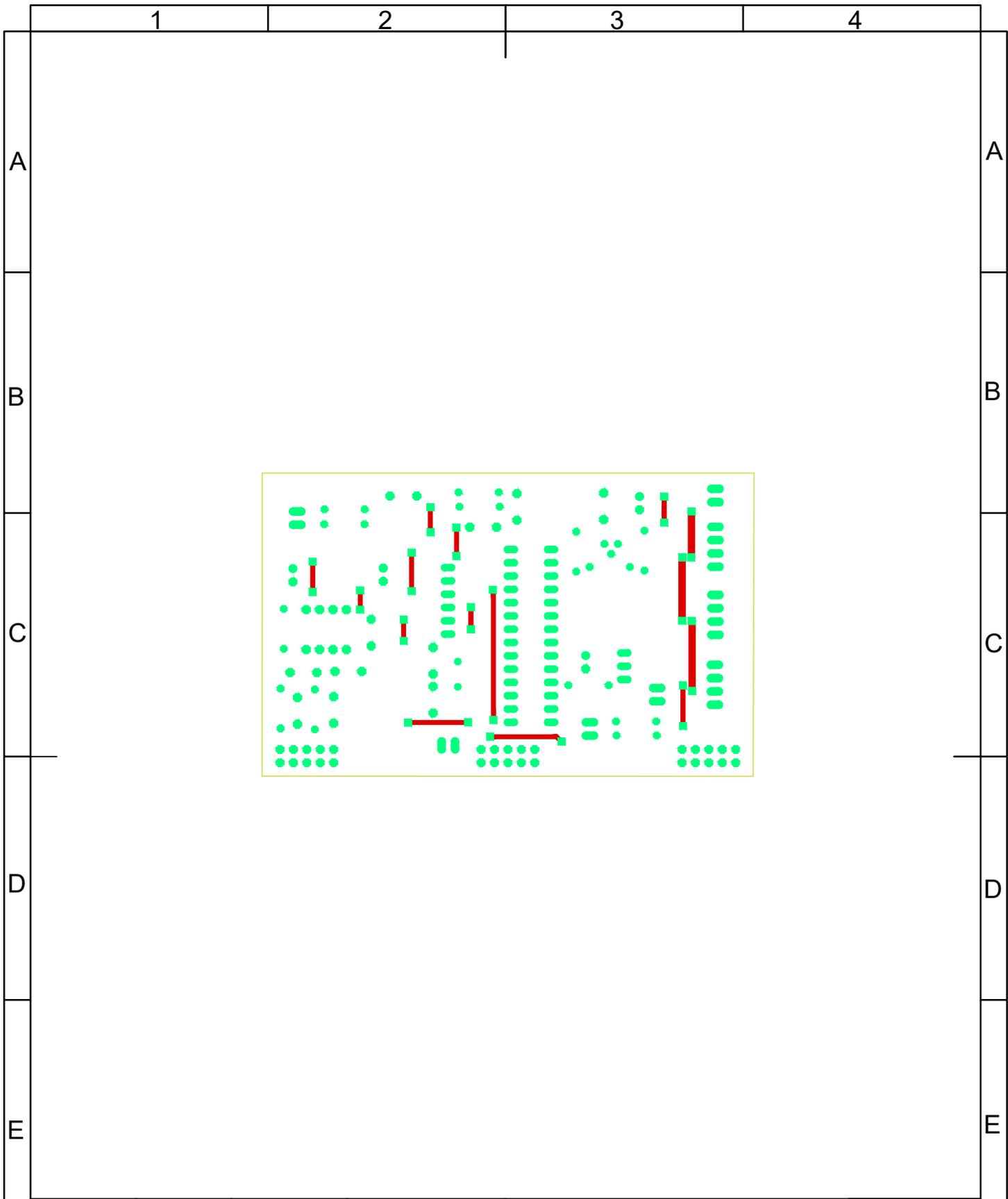


	Fecha	Nombre	Firma	ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL UNIVERSIDAD DE LA RIOJA Grado en ingeniería electrónica industrial y automática	
Dibujado	21/08/19	Alberto Otero			
Comprobado	21/08/19	Alberto Otero			
I.d.s.normas					
Escala	Controlador DC - DC ZVS			Número	20
Sin Escala	Esquema eléctrico			REFERENCIA:	
				Sustituye a	
				Sustituido por	

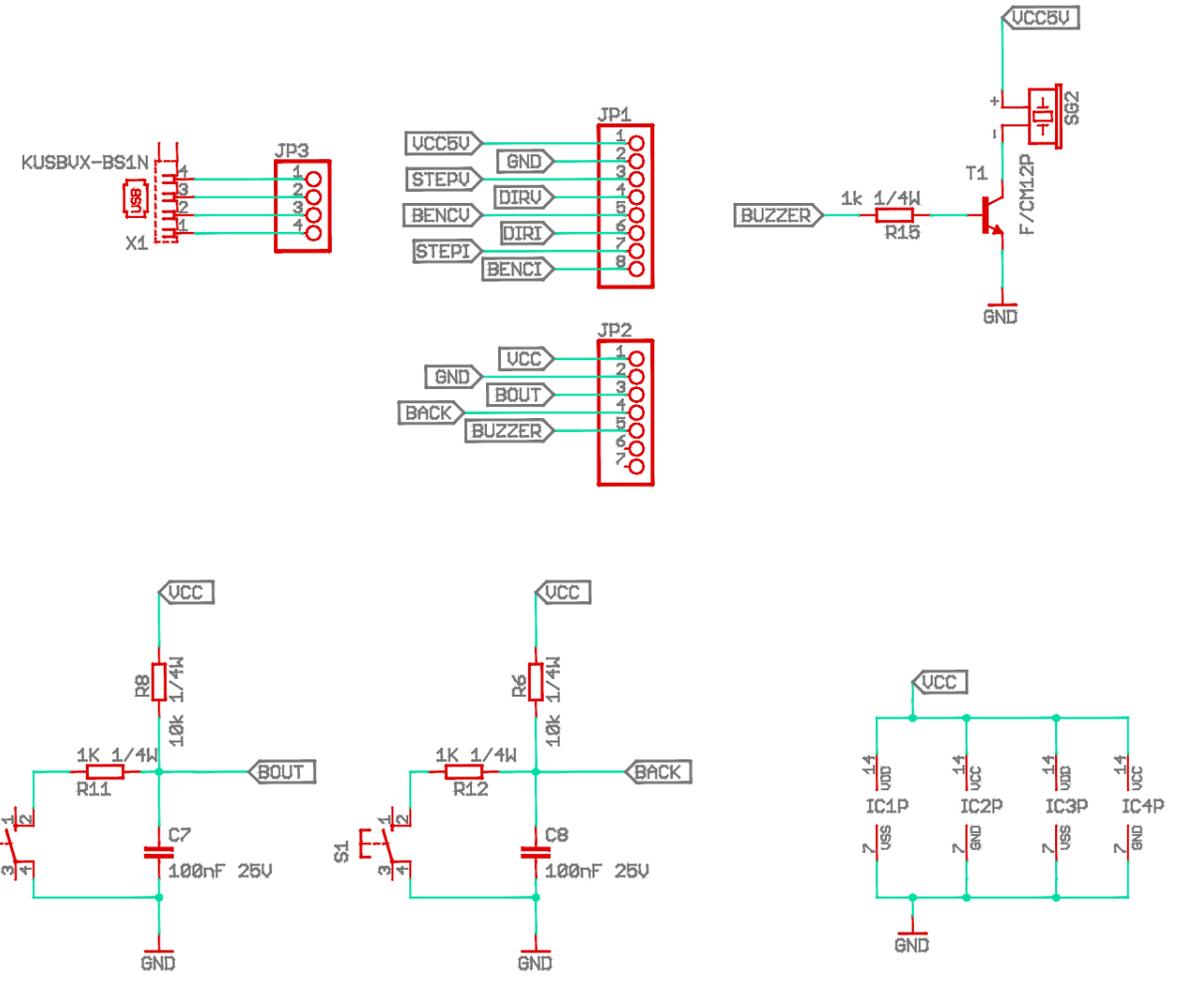
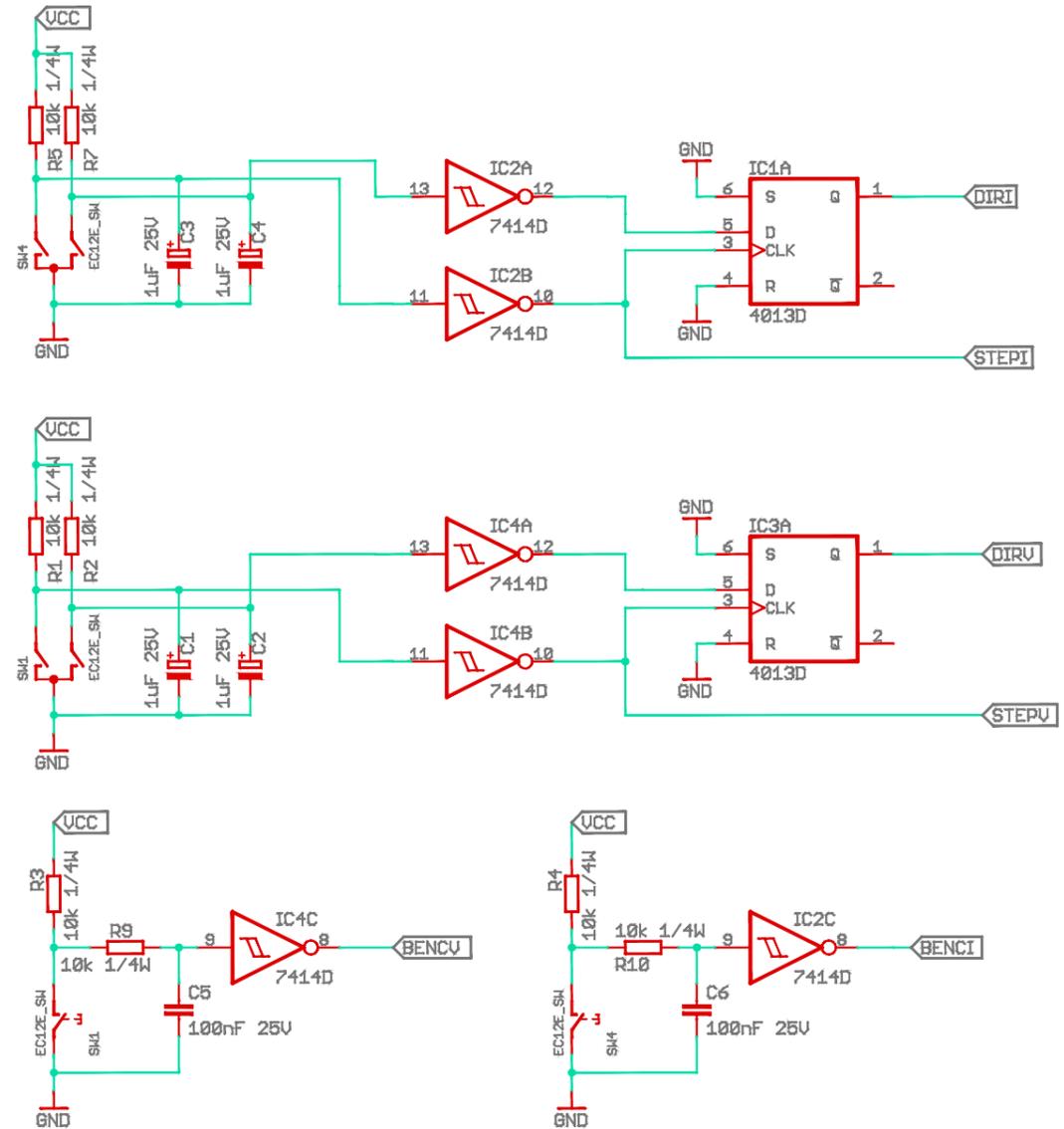






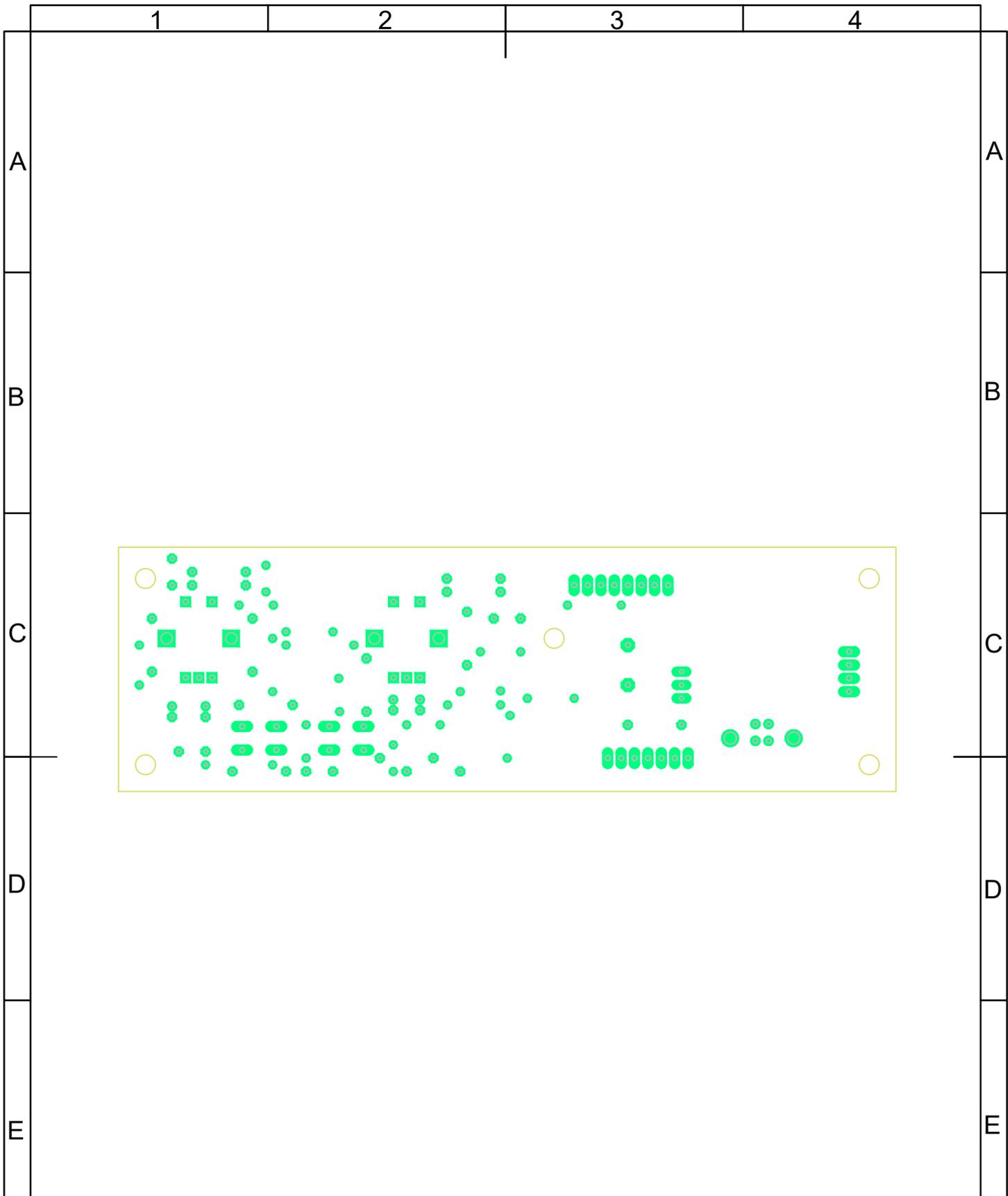


	Fecha	Nombre	Firma	ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL <b>UNIVERSIDAD DE LA RIOJA</b> <small>Grado en ingeniería electrónica industrial y automática</small>	
Dibujado	21/08/19	Alberto Otero			
Comprobado	21/08/19	Alberto Otero			
I.d.s.normas					
F	Escala 1:1	Controlador DC - DC ZVS		Número 24	F
Pistas - Capa superior				REFERENCIA:	
				Sustituye a	
				Sustituido por	



	Fecha	Nombre	Firma	ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL UNIVERSIDAD DE LA RIOJA Grado en ingeniería electrónica industrial y automática	
Dibujado	21/08/19	Alberto Otero			
Comprobado	21/08/19	Alberto Otero			
I.d.s.normas					
Escales	Botonera y puerto USB			Número	25
Sin Escala	Esquema eléctrico			REFERENCIA:	
				Sustituye a	
				Sustituido por	



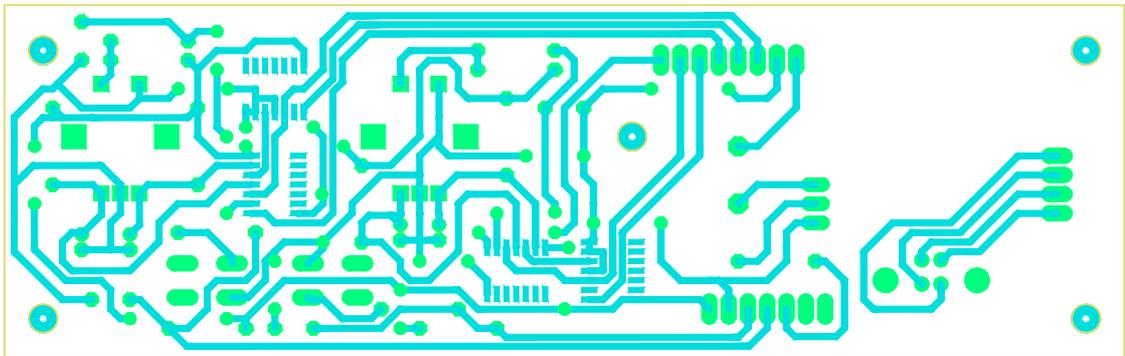


	Fecha	Nombre	Firma
Dibujado	21/08/19	Alberto Otero	
Comprobado	21/08/19	Alberto Otero	
I.d.s.normas			

ESCUELA TÉCNICA SUPERIOR  
 DE INGENIERÍA INDUSTRIAL  
**UNIVERSIDAD DE LA RIOJA**  
 Grado en ingeniería electrónica industrial y automática



Escala 1:1	Botonera y puerto USB	Número 27
	Layout Taladros	REFERENCIA:
		Sustituye a
		Sustituido por



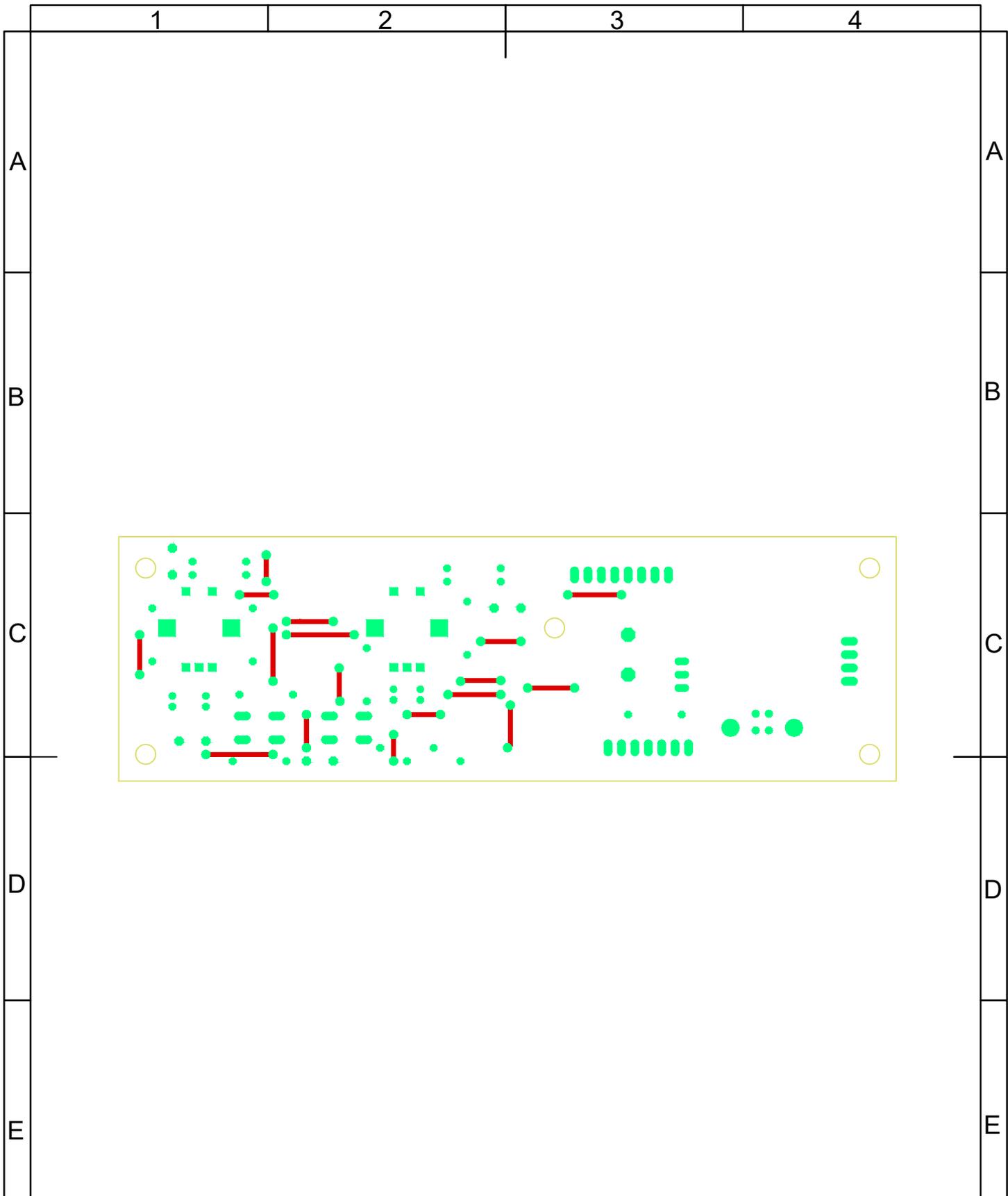
	Fecha	Nombre	Firma
Dibujado	21/08/19	Alberto Otero	
Comprobado	21/08/19	Alberto Otero	
I.d.s.normas			

ESCUELA TÉCNICA SUPERIOR  
 DE INGENIERÍA INDUSTRIAL  
**UNIVERSIDAD DE LA RIOJA**  
 Grado en ingeniería electrónica industrial y automática

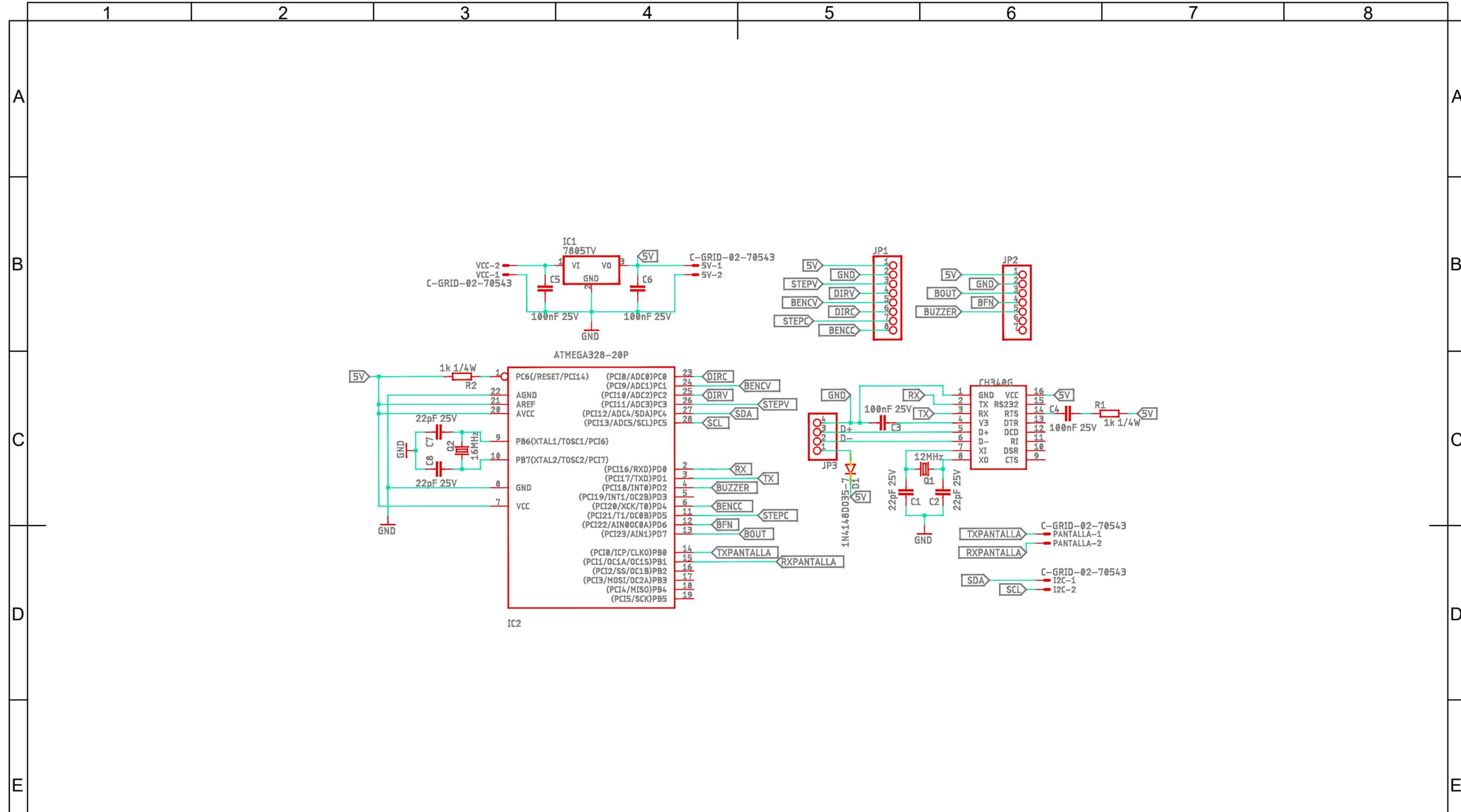


Escala 1:1	Botonera y puerto USB	Número 28
	Pistas - Capa inferior	REFERENCIA:
		Sustituye a
		Sustituido por





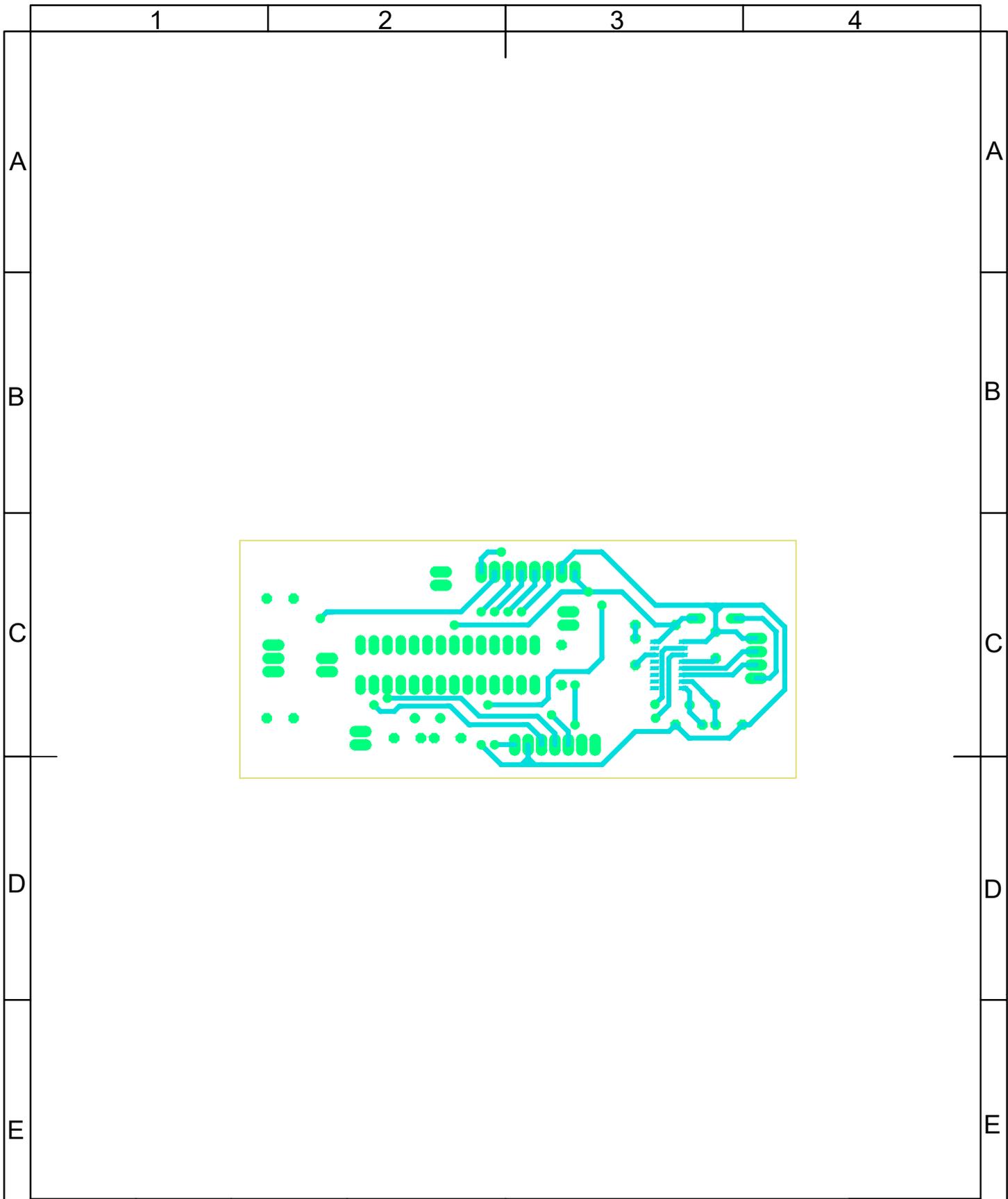
	Fecha	Nombre	Firma	ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL <b>UNIVERSIDAD DE LA RIOJA</b> <small>Grado en ingeniería electrónica industrial y automática</small>	
Dibujado	21/08/19	Alberto Otero			
Comprobado	21/08/19	Alberto Otero			
I.d.s.normas					
Escala 1:1	Botonera y puerto USB			Número 29	
	Pistas - Capa superior			REFERENCIA:	
				Sustituye a	
			Sustituido por		



	Fecha	Nombre	Firma	ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL UNIVERSIDAD DE LA RIOJA Grado en ingeniería electrónica industrial y automática	
Dibujado	21/08/19	Alberto Otero			
Comprobado	21/08/19	Alberto Otero			
I.d.s.normas					
Escala	Controlador panel frontal			Número	30
Sin Escala	Esquema eléctrico			REFERENCIA:	
				Sustituye a	
				Sustituido por	





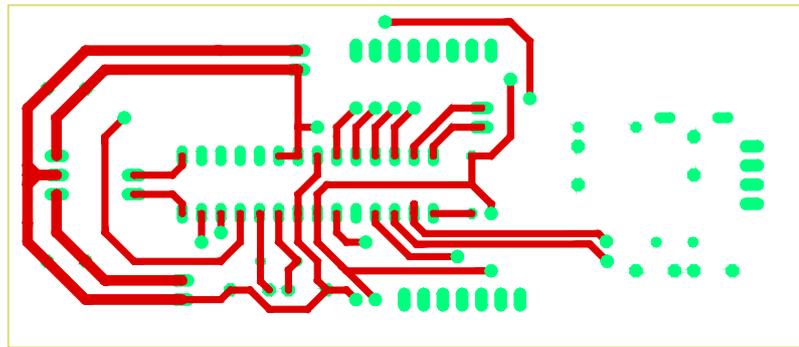


	Fecha	Nombre	Firma
Dibujado	21/08/19	Alberto Otero	
Comprobado	21/08/19	Alberto Otero	
I.d.s.normas			

ESCUELA TÉCNICA SUPERIOR  
 DE INGENIERÍA INDUSTRIAL  
**UNIVERSIDAD DE LA RIOJA**  
 Grado en ingeniería electrónica industrial y automática



<b>F</b> Escala 1:1	Controlador Panel Frontal	Número 33
	Pistas - Capa inferior	REFERENCIA: Sustituye a
		Sustituido por

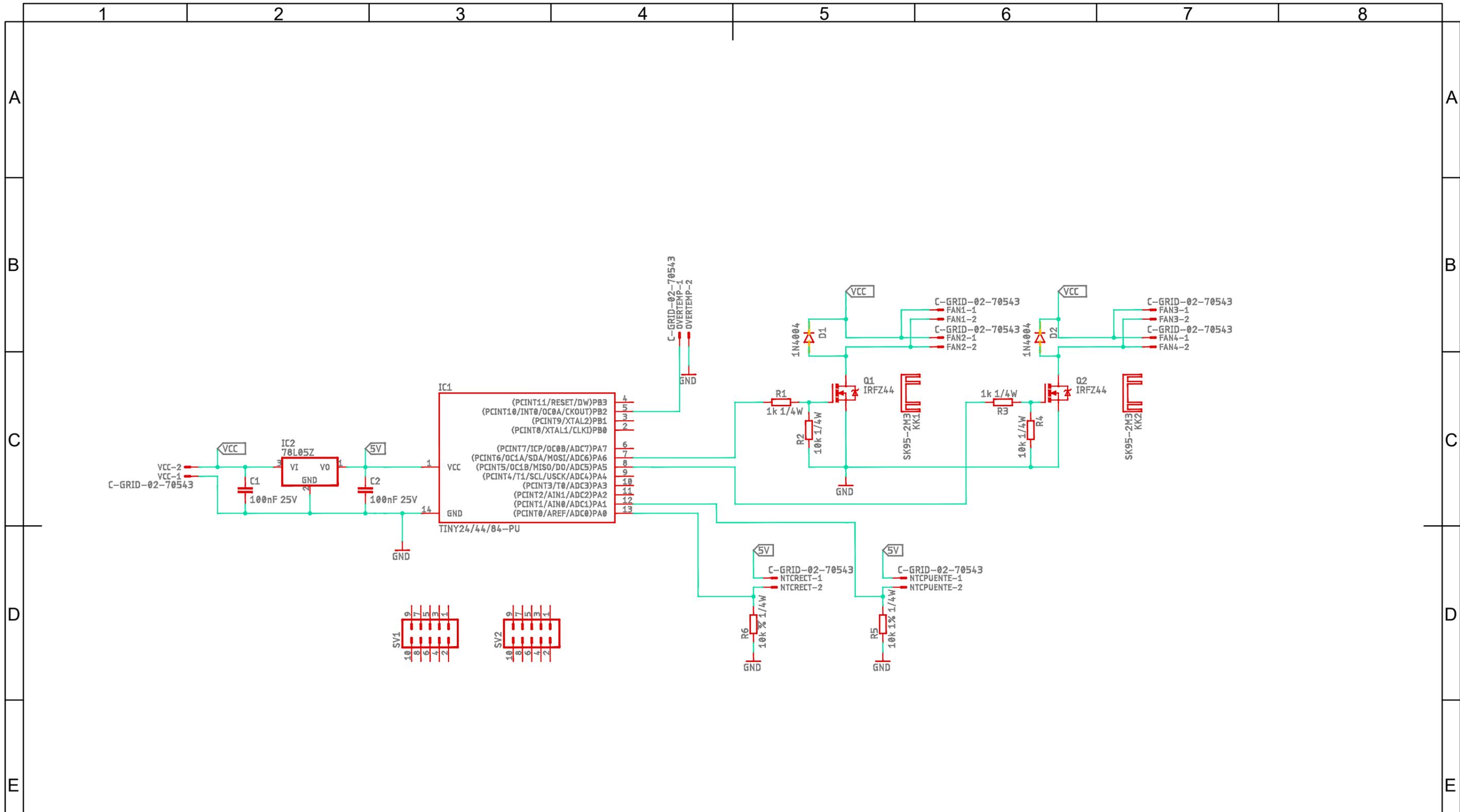


	Fecha	Nombre	Firma
Dibujado	21/08/19	Alberto Otero	
Comprobado	21/08/19	Alberto Otero	
I.d.s.normas			

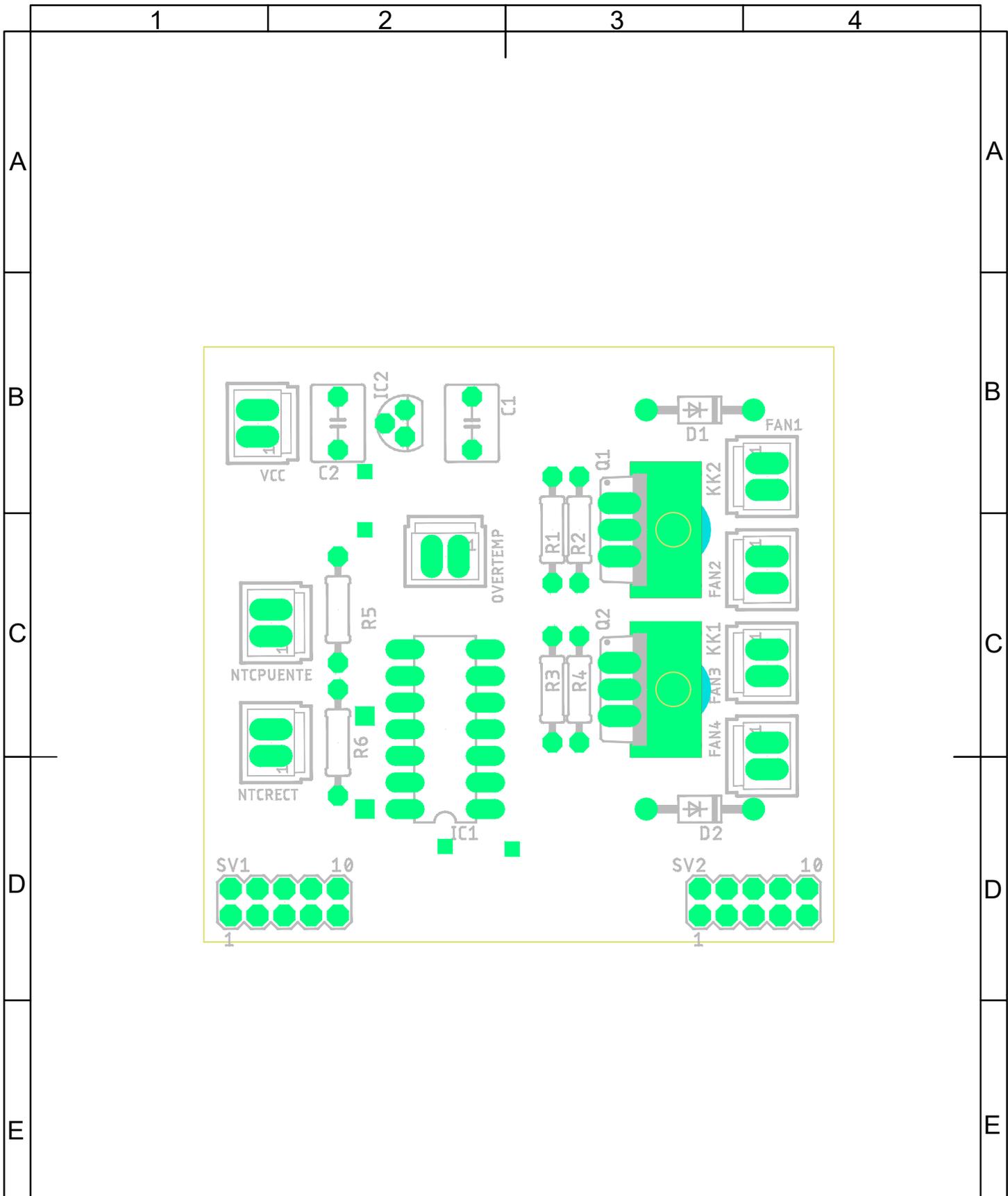
ESCUELA TÉCNICA SUPERIOR  
 DE INGENIERÍA INDUSTRIAL  
 UNIVERSIDAD DE LA RIOJA  
 Grado en ingeniería electrónica industrial y automática



F	Escala 1:1	Controlador Panel Frontal	Número 34	F
		Pistas - Capa superior	REFERENCIA:	
			Sustituye a	
	Sustituido por			



	Fecha	Nombre	Firma	ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL UNIVERSIDAD DE LA RIOJA Grado en ingeniería electrónica industrial y automática	
Dibujado	21/08/19	Alberto Otero			
Comprobado	21/08/19	Alberto Otero			
I.d.s.normas					
Escala	Controlador de temperatura			Número	35
Sin Escala	Esquema eléctrico			REFERENCIA:	
				Sustituye a	
				Sustituido por	

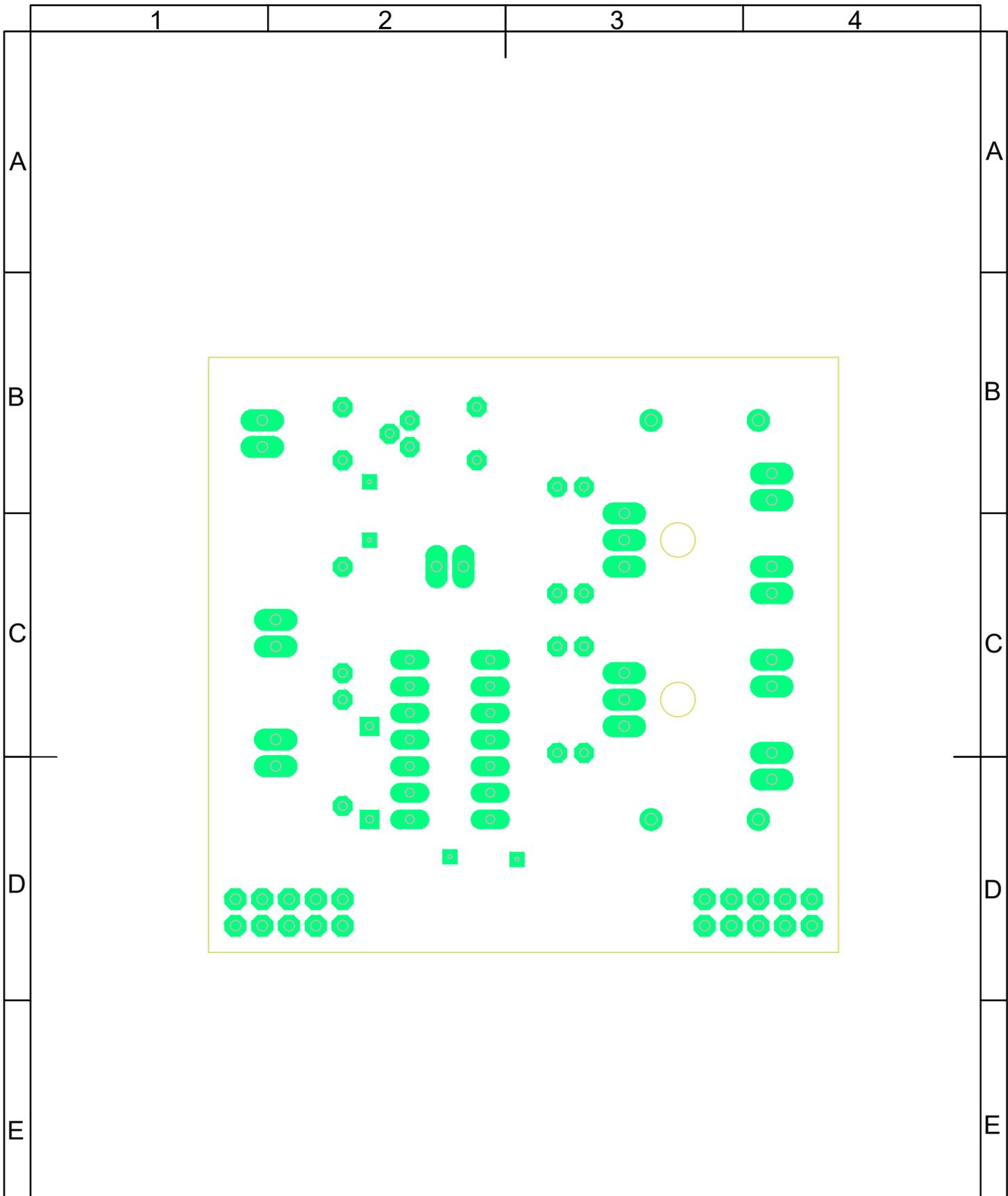


	Fecha	Nombre	Firma
Dibujado	21/08/19	Alberto Otero	
Comprobado	21/08/19	Alberto Otero	
I.d.s.normas			

ESCUELA TÉCNICA SUPERIOR  
 DE INGENIERÍA INDUSTRIAL  
 UNIVERSIDAD DE LA RIOJA  
 Grado en ingeniería electrónica industrial y automática



Escala 2:1	Controlador de temperatura	Número 36
	Layout componentes	REFERENCIA:
		Sustituye a
		Sustituido por

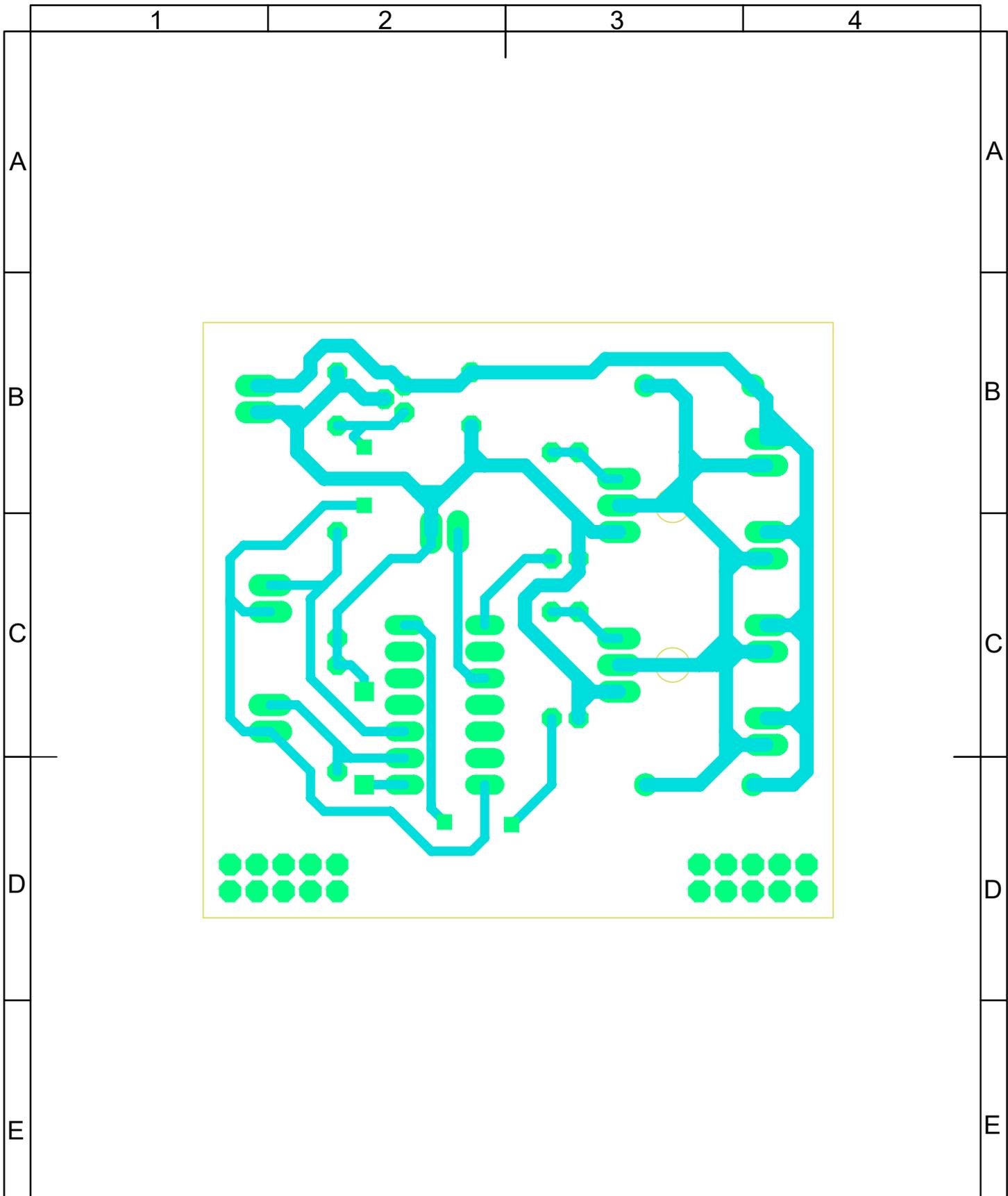


	Fecha	Nombre	Firma
Dibujado	21/08/19	Alberto Otero	
Comprobado	21/08/19	Alberto Otero	
I.d.s.normas			

ESCUELA TÉCNICA SUPERIOR  
 DE INGENIERÍA INDUSTRIAL  
**UNIVERSIDAD DE LA RIOJA**  
 Grado en ingeniería electrónica industrial y automática



Escala 2:1	Controlador de temperatura	Número 37
	Layout Taladros	REFERENCIA:
		Sustituye a
		Sustituido por

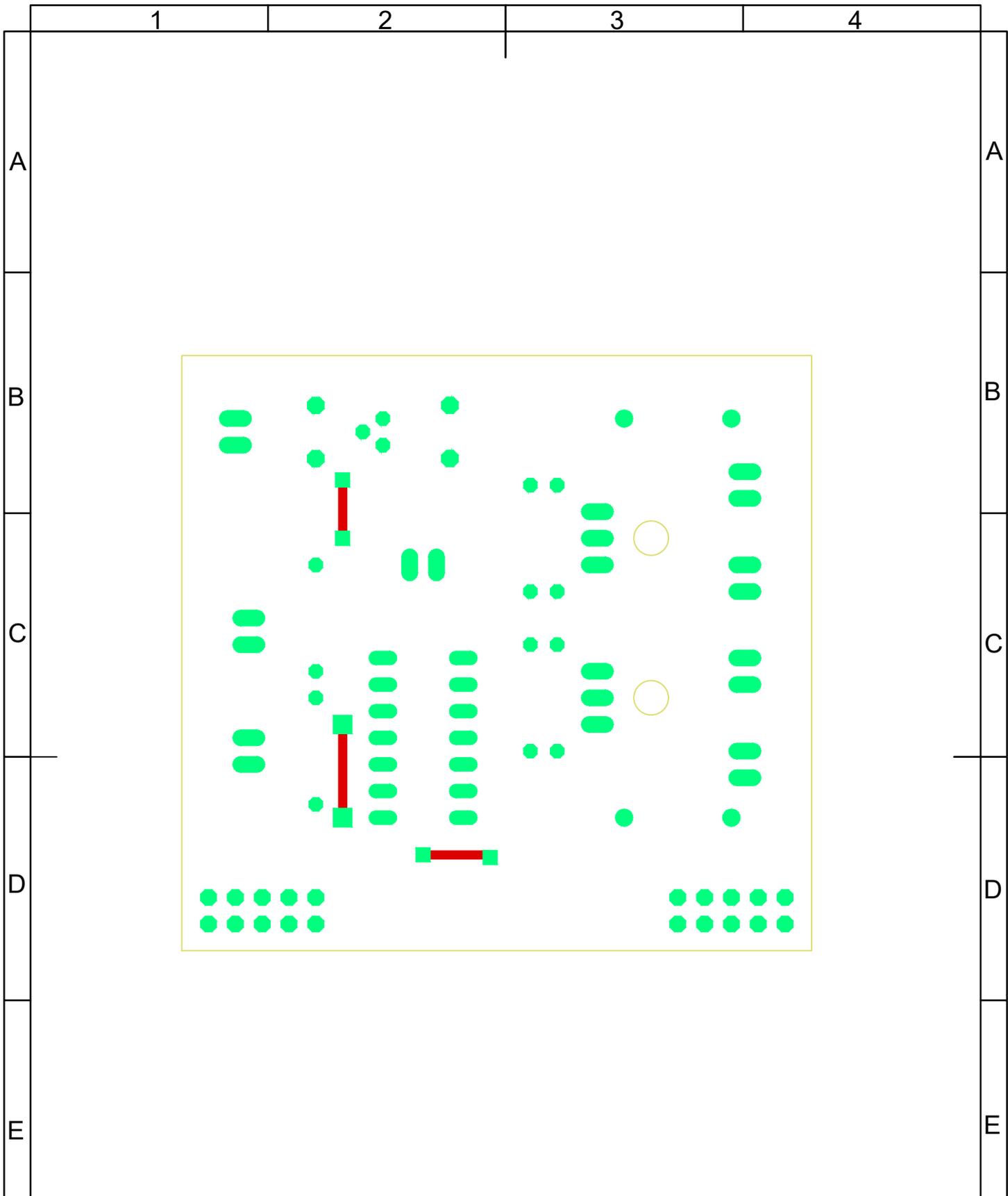


	Fecha	Nombre	Firma
Dibujado	21/08/19	Alberto Otero	
Comprobado	21/08/19	Alberto Otero	
I.d.s.normas			

ESCUELA TÉCNICA SUPERIOR  
 DE INGENIERÍA INDUSTRIAL  
**UNIVERSIDAD DE LA RIOJA**  
 Grado en ingeniería electrónica industrial y automática



Escala 2:1	Controlador de temperatura	Número 38
	Pistas - Capa inferior	REFERENCIA:
		Sustituye a
		Sustituido por

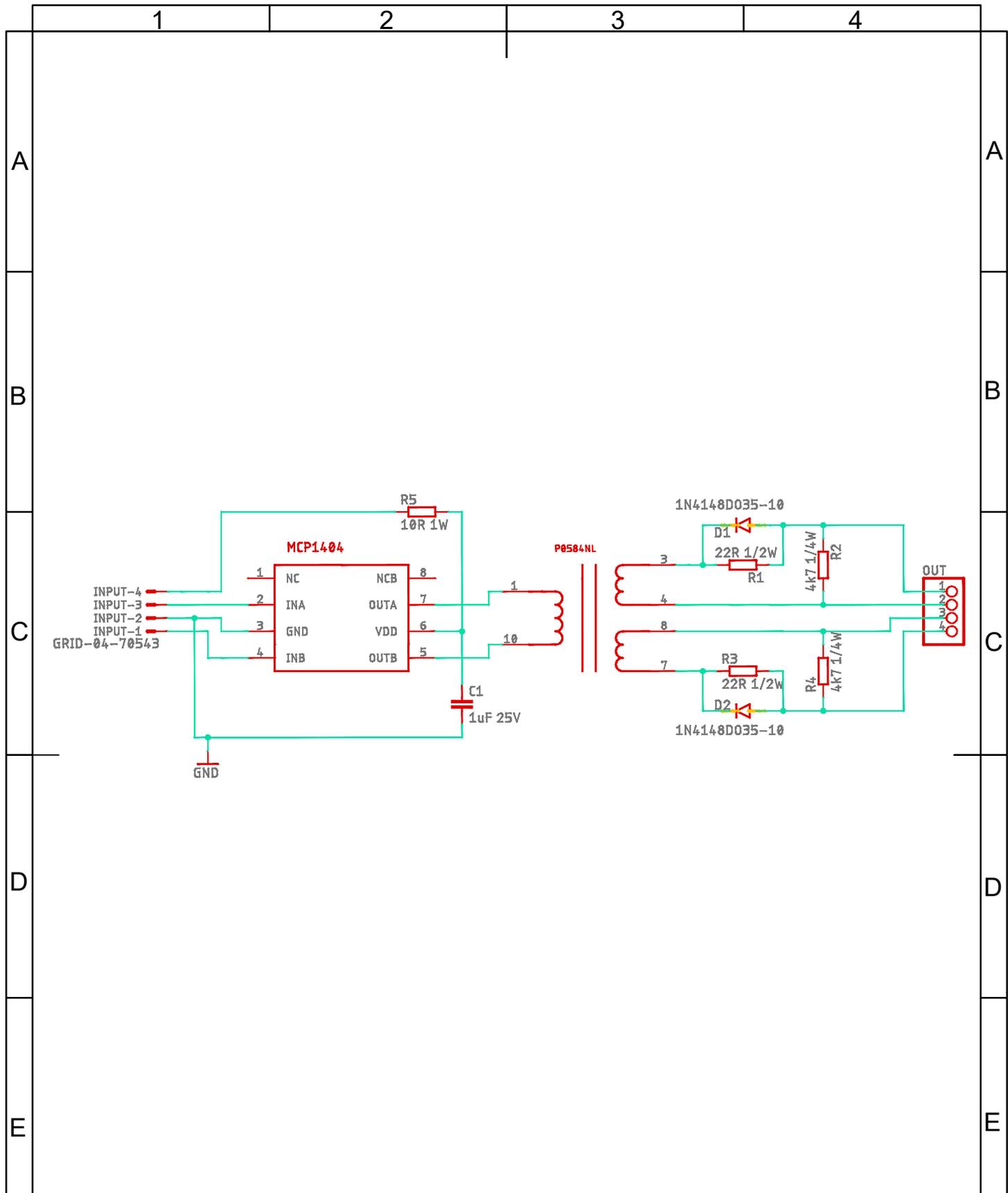


	Fecha	Nombre	Firma
Dibujado	21/08/19	Alberto Otero	
Comprobado	21/08/19	Alberto Otero	
I.d.s.normas			

ESCUELA TÉCNICA SUPERIOR  
 DE INGENIERÍA INDUSTRIAL  
 UNIVERSIDAD DE LA RIOJA  
 Grado en ingeniería electrónica industrial y automática



Escala 2:1	Controlador de temperatura	Número 39
	Pistas - Capa superior	REFERENCIA:
		Sustituye a
		Sustituido por

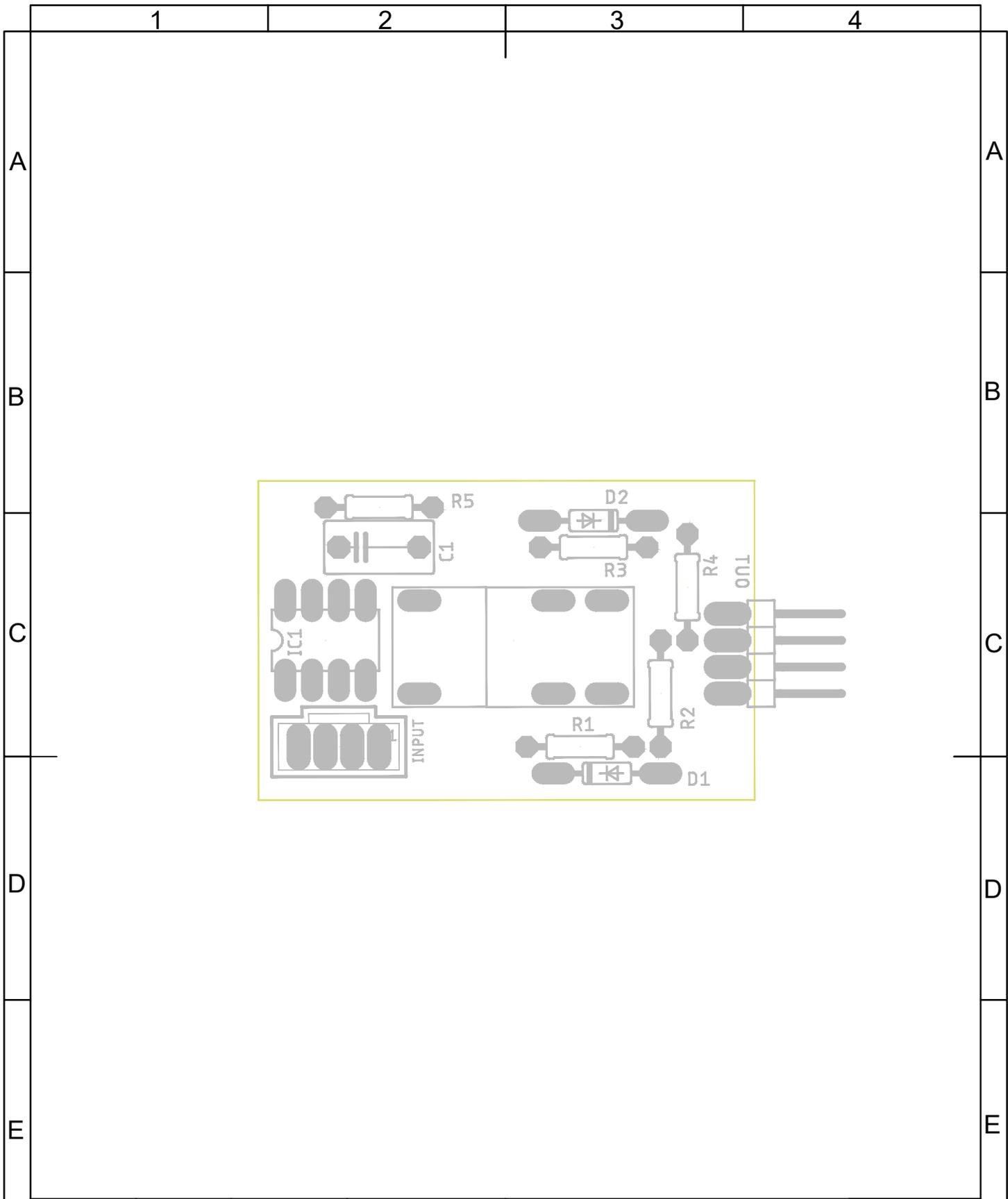


	Fecha	Nombre	Firma
Dibujado	21/08/19	Alberto Otero	
Comprobado	21/08/19	Alberto Otero	
I.d.s.normas			

ESCUELA TÉCNICA SUPERIOR  
 DE INGENIERÍA INDUSTRIAL  
 UNIVERSIDAD DE LA RIOJA  
 Grado en ingeniería electrónica industrial y automática

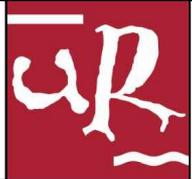


Escala Sin Escala	Driver Puente completo	Número 40
	Esquema eléctrico	REFERENCIA:
		Sustituye a
		Sustituido por

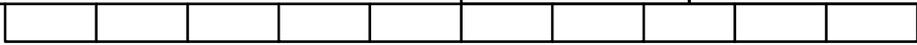


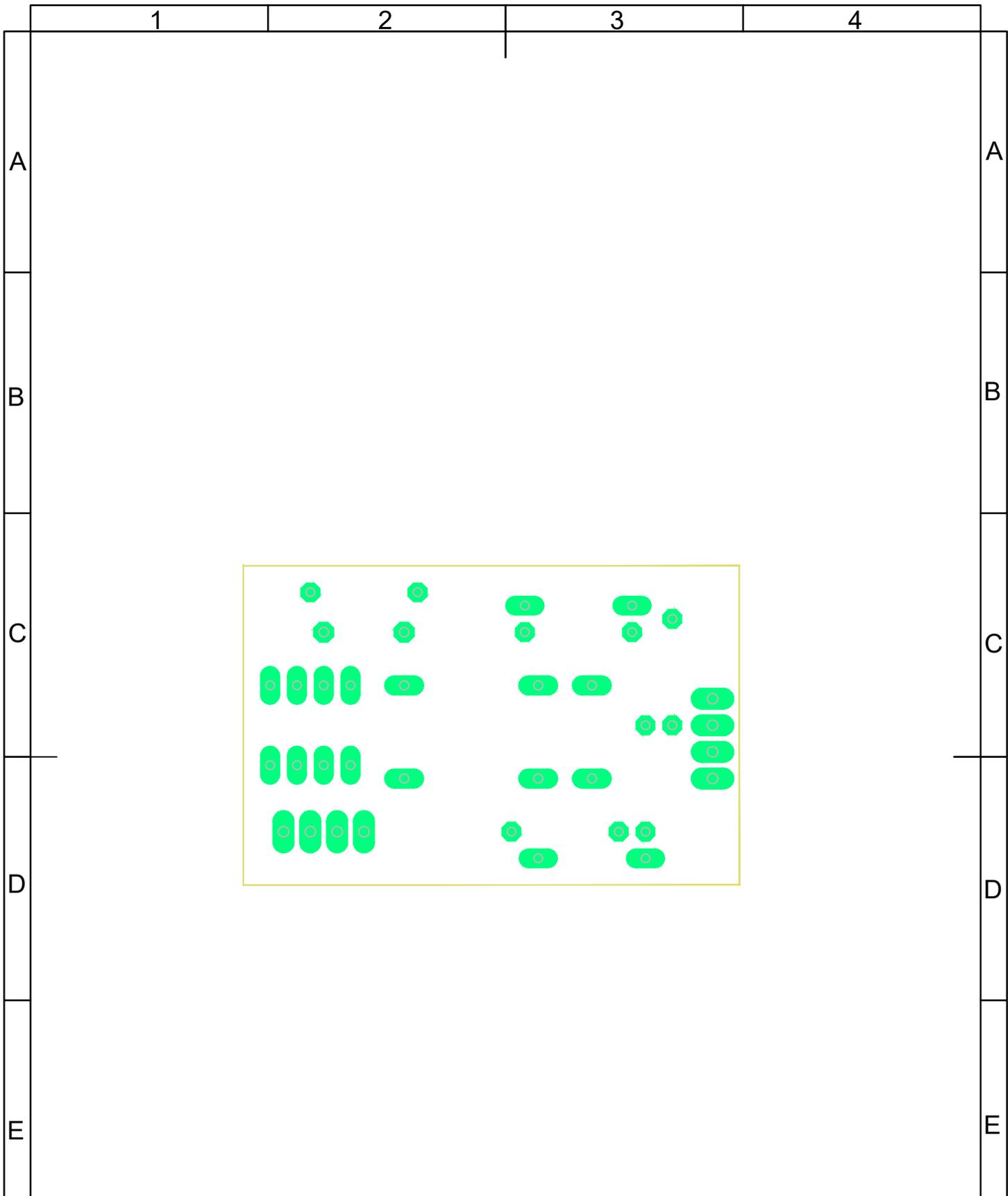
	Fecha	Nombre	Firma
Dibujado	21/08/19	Alberto Otero	
Comprobado	21/08/19	Alberto Otero	
I.d.s.normas			

**ESCUELA TÉCNICA SUPERIOR  
 DE INGENIERÍA INDUSTRIAL**  
**UNIVERSIDAD DE LA RIOJA**  
 Grado en ingeniería electrónica industrial y automática

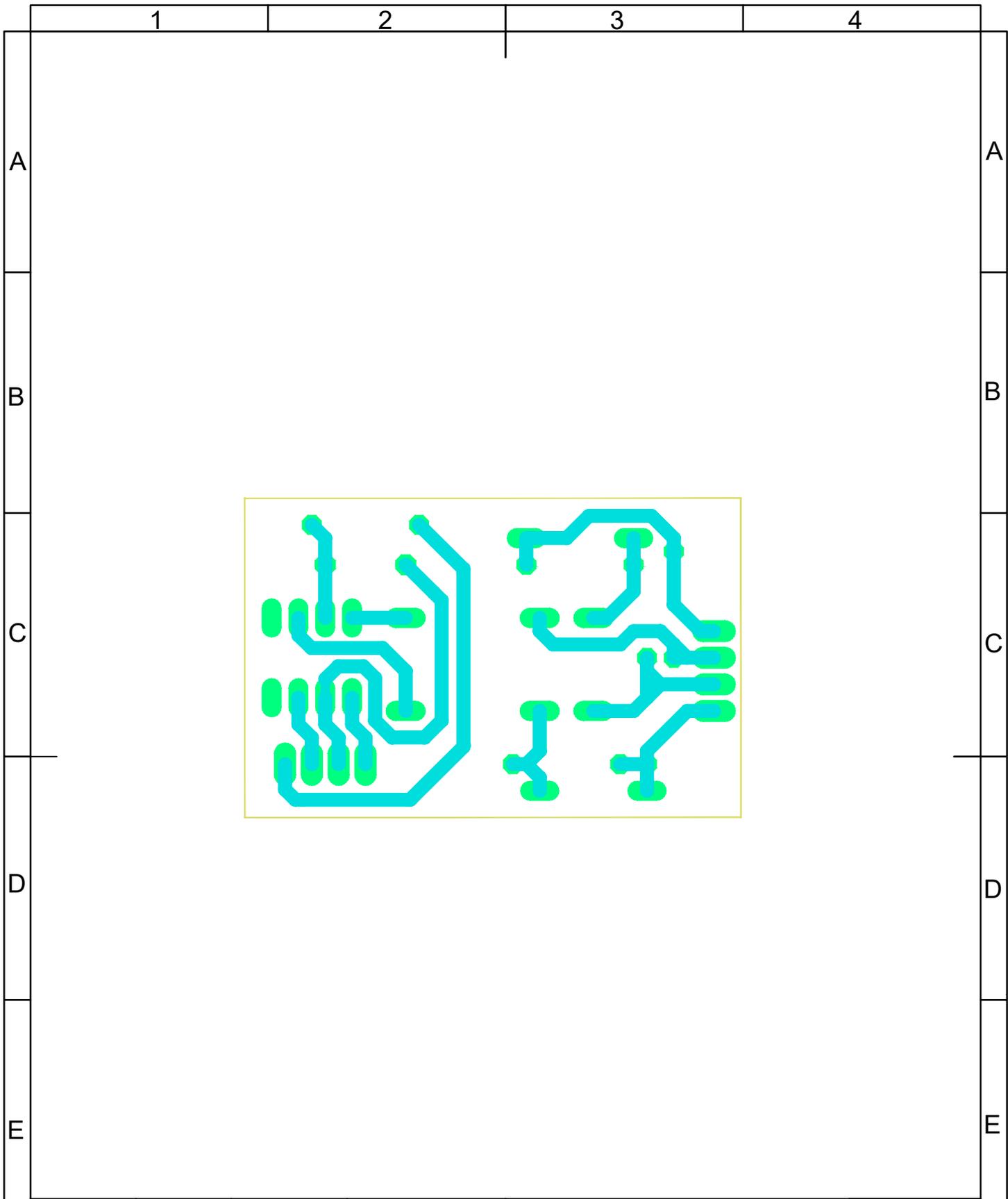


<b>F</b> Escala 2:1	Driver Puente completo	Número 41
	Layout componentes	<b>REFERENCIA:</b>
		Sustituye a Sustituido por



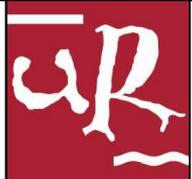


	Fecha	Nombre	Firma	ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL <b>UNIVERSIDAD DE LA RIOJA</b> <small>Grado en ingeniería electrónica industrial y automática</small>	
Dibujado	21/08/19	Alberto Otero			
Comprobado	21/08/19	Alberto Otero			
I.d.s.normas					
Escala 2:1	Driver Puente Completo			Número 42	
	Layout Taladros			REFERENCIA:	
				Sustituye a	
			Sustituido por		

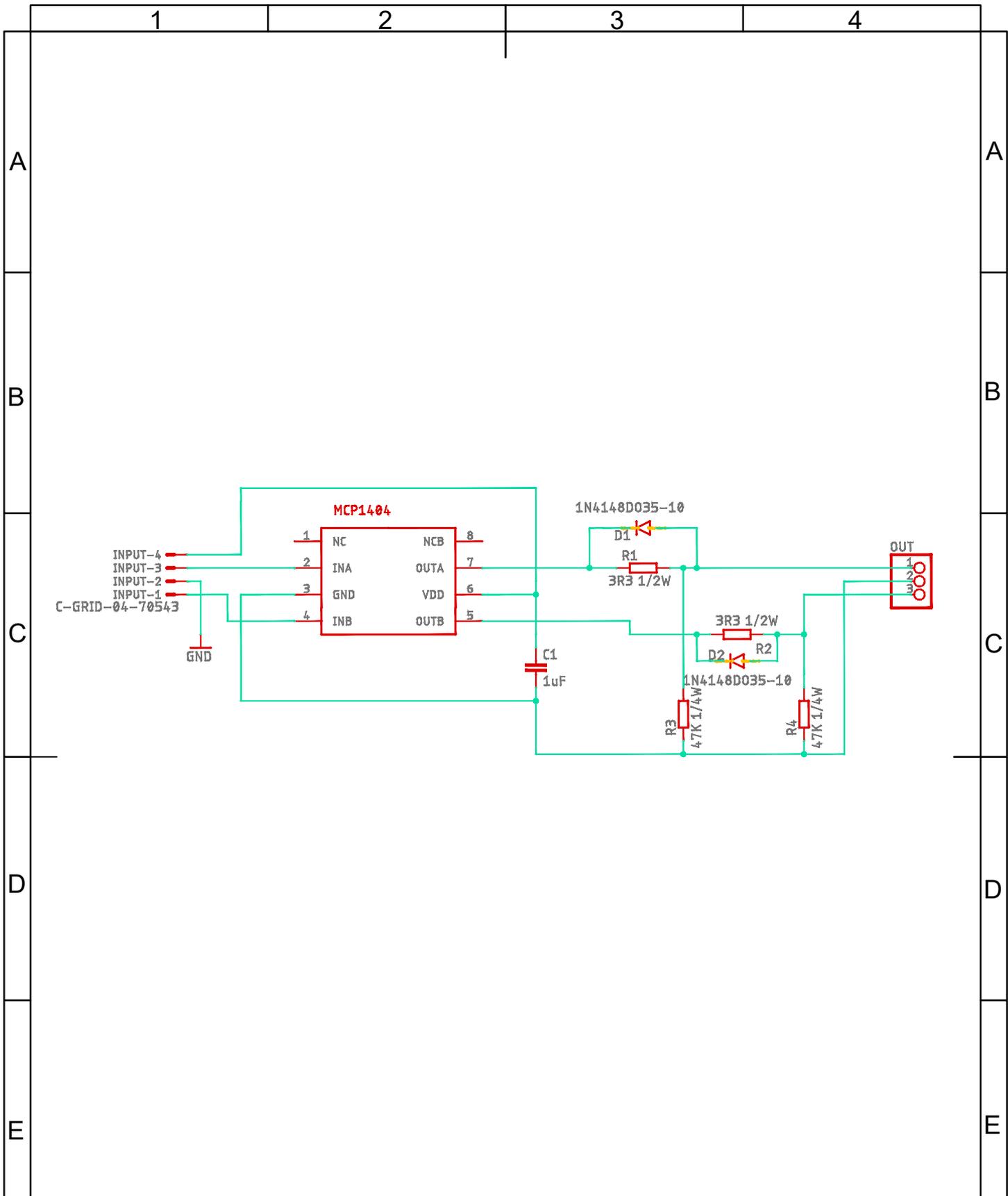


	Fecha	Nombre	Firma
Dibujado	21/08/19	Alberto Otero	
Comprobado	21/08/19	Alberto Otero	
I.d.s.normas			

ESCUELA TÉCNICA SUPERIOR  
 DE INGENIERÍA INDUSTRIAL  
**UNIVERSIDAD DE LA RIOJA**  
 Grado en ingeniería electrónica industrial y automática



F	Escala 2:1	Driver Puente Completo	Número 43	F
		Pistas - Capa inferior	REFERENCIA:	
			Sustituye a Sustituido por	

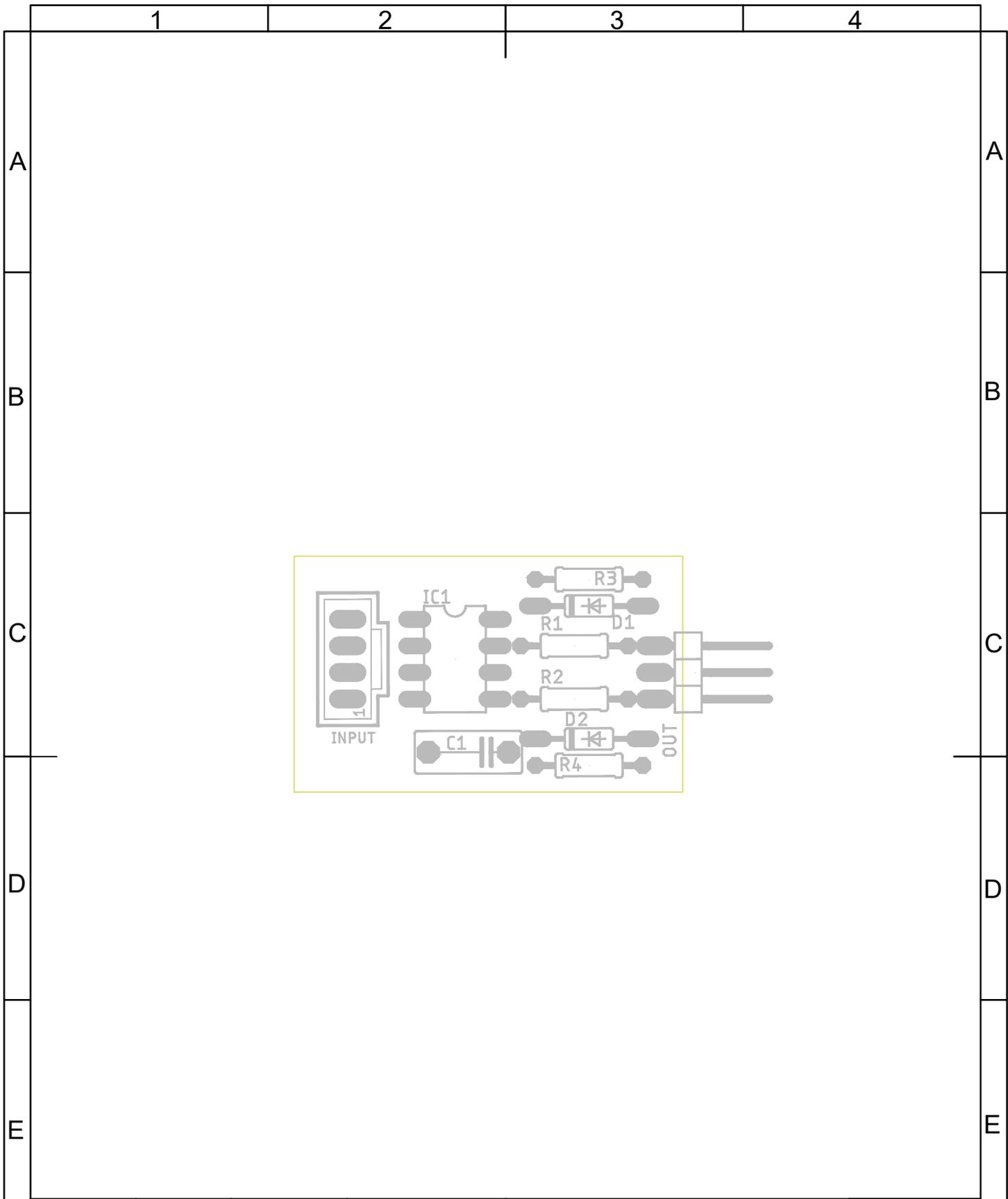


	Fecha	Nombre	Firma
Dibujado	21/08/19	Alberto Otero	
Comprobado	21/08/19	Alberto Otero	
I.d.s.normas			

**ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL**  
**UNIVERSIDAD DE LA RIOJA**  
 Grado en ingeniería electrónica industrial y automática



Escala Sin Escala	Driver rectificación síncrona	Número 44
	Esquema eléctrico	REFERENCIA:
		Sustituye a
		Sustituido por

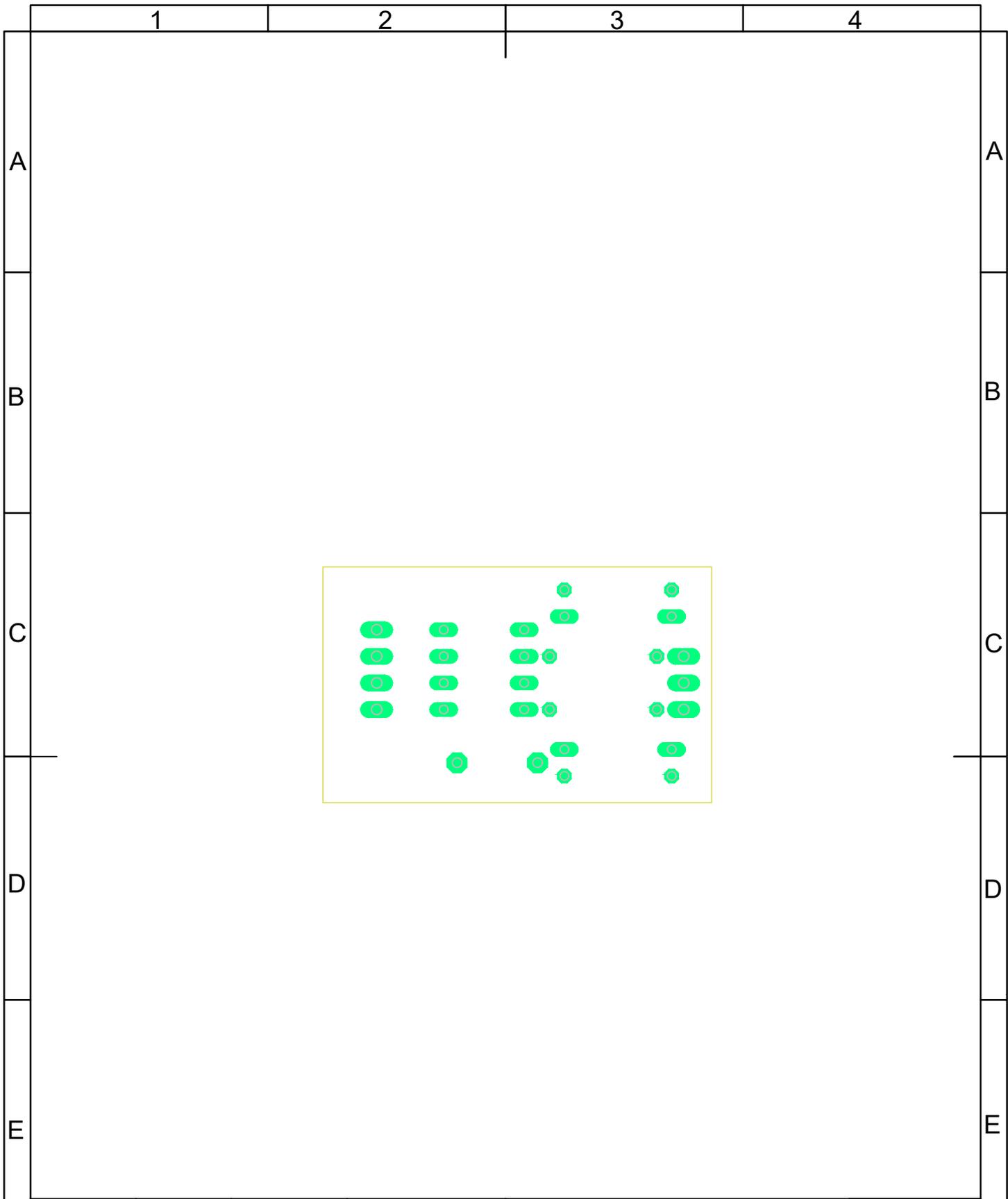


	Fecha	Nombre	Firma
Dibujado	21/08/19	Alberto Otero	
Comprobado	21/08/19	Alberto Otero	
I.d.s.normas			

ESCUELA TÉCNICA SUPERIOR  
 DE INGENIERÍA INDUSTRIAL  
 UNIVERSIDAD DE LA RIOJA  
 Grado en ingeniería electrónica industrial y automática

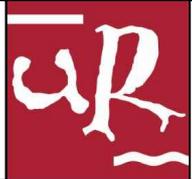


Escala 2:1	Driver rectificación síncrona	Número 45
	Layout componentes	REFERENCIA:
		Sustituye a
		Sustituido por

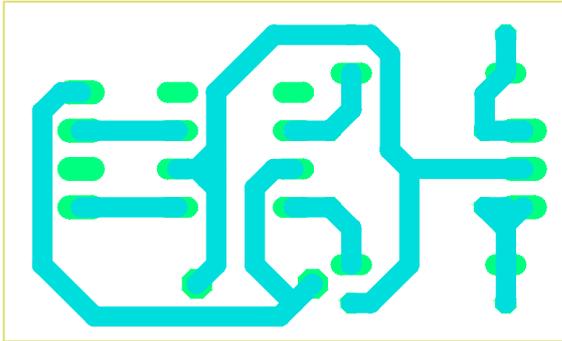
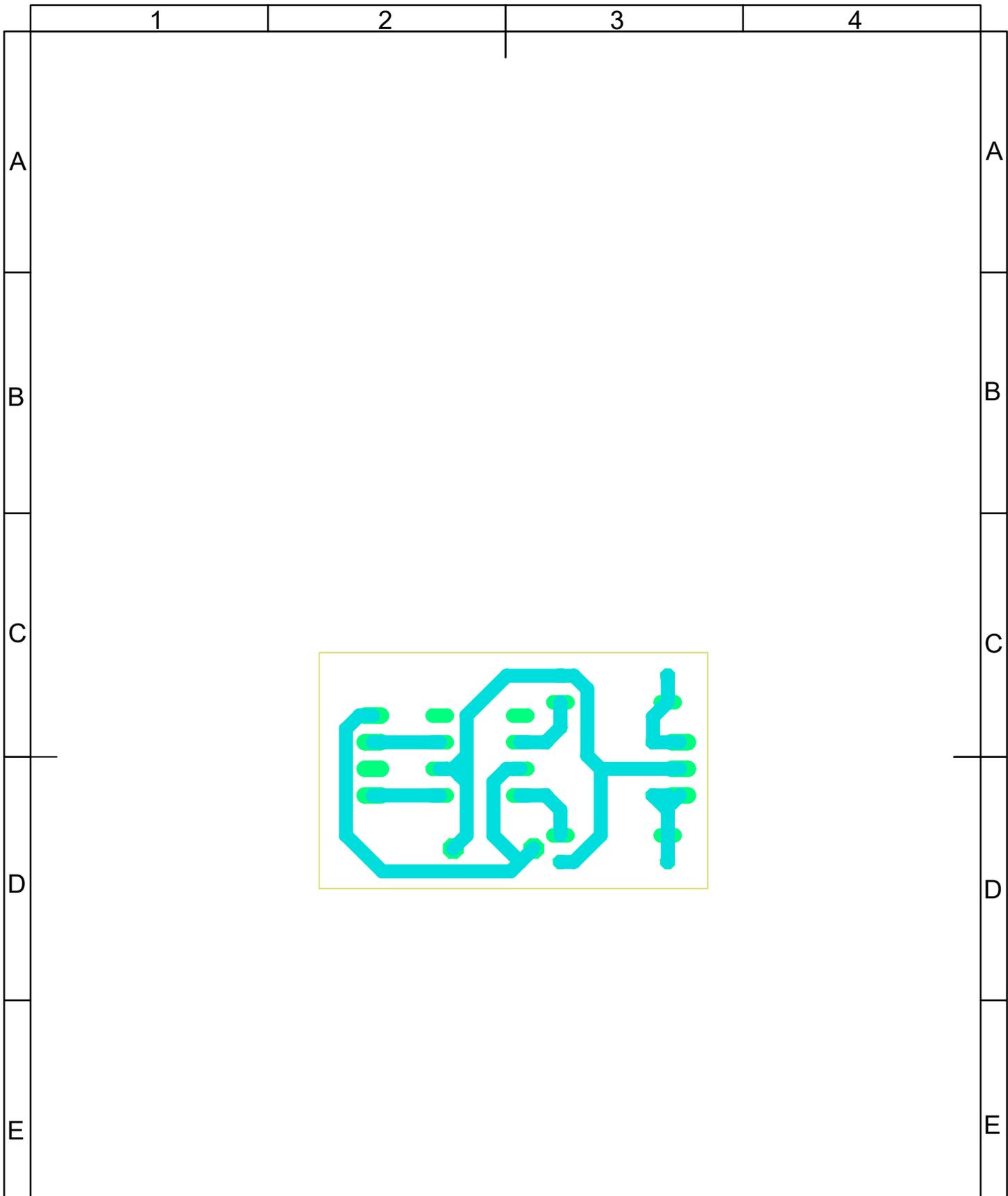


	Fecha	Nombre	Firma
Dibujado	21/08/19	Alberto Otero	
Comprobado	21/08/19	Alberto Otero	
I.d.s.normas			

**ESCUELA TÉCNICA SUPERIOR  
 DE INGENIERÍA INDUSTRIAL**  
**UNIVERSIDAD DE LA RIOJA**  
 Grado en ingeniería electrónica industrial y automática



Escala 2:1	Driver rectificación síncrona	Número 46
	Layout - Agujeros	REFERENCIA:
		Sustituye a
		Sustituido por



	Fecha	Nombre	Firma
Dibujado	21/08/19	Alberto Otero	
Comprobado	21/08/19	Alberto Otero	
I.d.s.normas			

**ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL**  
**UNIVERSIDAD DE LA RIOJA**  
 Grado en ingeniería electrónica industrial y automática

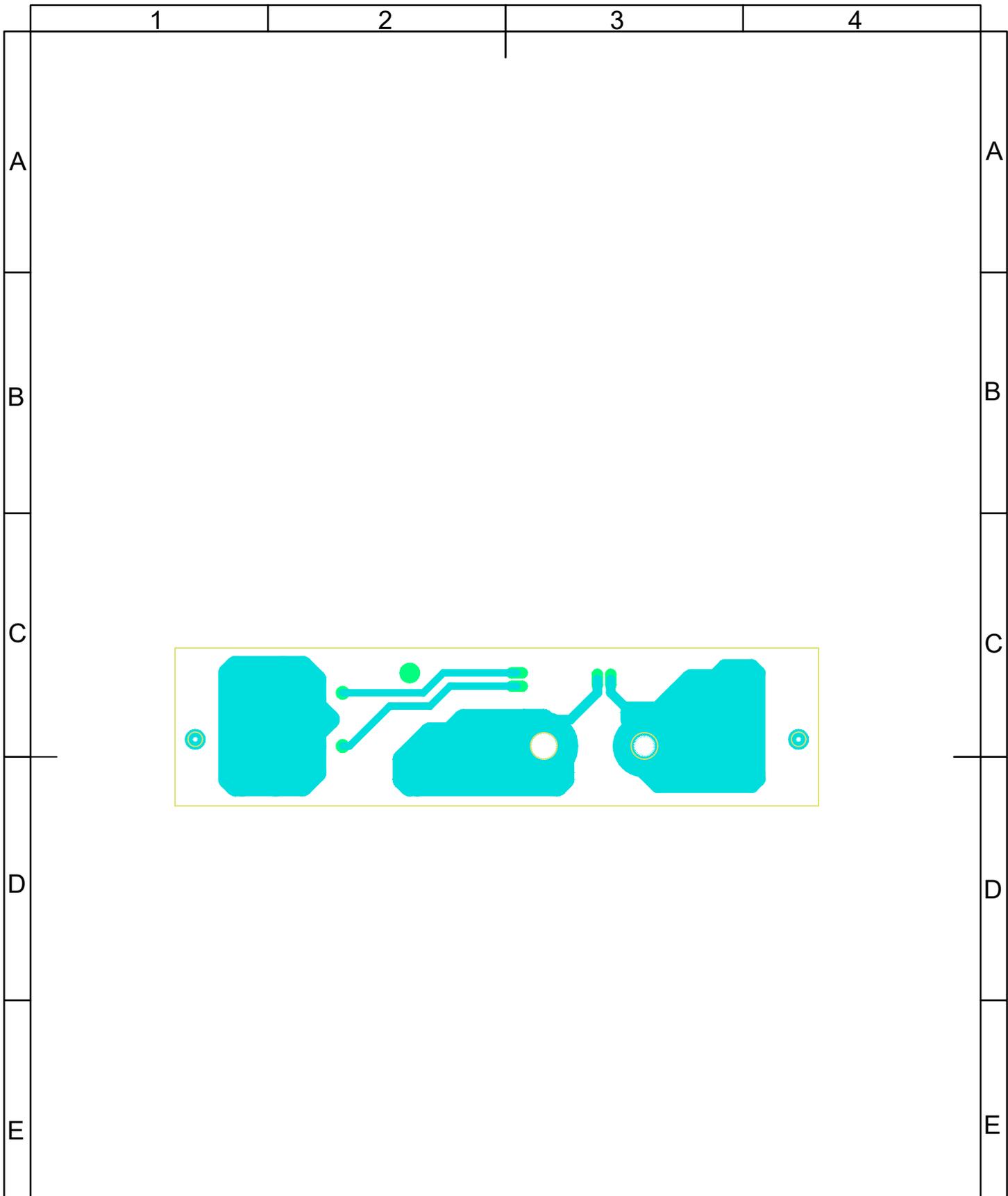


<b>F</b> Escala 2:1	Driver rectificación síncrona		Número 47
	Pistas - Capa inferior		REFERENCIA: Sustituye a
			Sustituido por









	Fecha	Nombre	Firma	ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL <b>UNIVERSIDAD DE LA RIOJA</b> Grado en ingeniería electrónica industrial y automática	
Dibujado	21/08/19	Alberto Otero			
Comprobado	21/08/19	Alberto Otero			
I.d.s.normas					
F	Escala 1:1	Etapa de salida			Número 51
		Pistas - Capa inferior			REFERENCIA:
					Sustituye a
			Sustituido por		



# PLIEGO DE PRESCRIPCIONES TÉCNICAS



## Índice

1. Objeto y ámbito del pliego.....	339
2. Definición de parámetros mínimos y máximos de operación.....	339
2.1. Filtro EMI y elementos de protección.....	339
2.2. Convertidor PFC.....	340
2.3. Convertidor DC – DC.....	340
2.4. Controlador de temperatura.....	341
2.5. Fuentes de alimentación auxiliares.....	341
2.6. Controlador DC – DC.....	342
2.7. Drivers de puente completo.....	342
2.8. Drivers de rectificación síncrona.....	342
2.9. Conectores internos del dispositivo.....	342
2.10. Terminal de entrada CA e interruptor.....	342
2.11. Cableado interno.....	343
2.12. Temperaturas máximas de operación.....	343
3. Parámetros de los protocolos de comunicaciones.....	343
3.1. Parámetros UART.....	343
3.2. Parámetros I2C.....	344
4. Parámetros recomendados de las piezas mecánicas.....	344
4.1. Fabricación de piezas mediante impresión 3D.....	344
4.2. Propiedades de la carcasa.....	345
5. Software.....	345



## 1. Objeto y ámbito del pliego

El presente pliego de prescripciones técnicas, nace con el objeto de definir los parámetros bajo los que se garantiza el correcto funcionamiento de la solución técnica descrita anteriormente y definir el uso correcto del dispositivo.

Además, se definen los requisitos mínimos que se han de cumplir para el dimensionamiento de los componentes considerados relevantes para la reproducción y/o reemplazo y reparación del sistema.

## 2. Definición de parámetros mínimos y máximos de operación

Se definen aquí, las prestaciones máximas que el dispositivo es capaz de ofrecer, y las condiciones mínimas y máximas que se deben respetar para su correcto funcionamiento.

### 2.1. Filtro EMI y elementos de protección

En esta etapa de la fuente se deben incluir los siguientes elementos de protección del esquema eléctrico del diseño original:

- Fusible HRC 250 VAC 10A 38 mm.
- Varistor 275 VAC.

En caso de sustitución deben seguir asegurando las prestaciones de los elementos. Por ejemplo, en el caso del fusible, deberá asegurarse que los tiempos de actuación sean iguales o más rápidos que el del elemento mencionado y que las tensiones de trabajo/corte sean iguales o superiores.

El modo de operación normal, ha sido establecido en base a una red eléctrica de 230 VAC y 50 Hz. Todos los componentes situados en serie con fase o neutro incluidos en esta etapa, deben tener una capacidad de corriente mínima de 10A. Los condensadores de filtrado del ruido en modo diferencial, deben estar garantizados con certificación X, y los de modo común con certificación Y, además, deberán soportar tensiones de 275 VAC o mayores.

Se establecen unas frecuencias de corte de obligado cumplimiento, que son:

- Frecuencia de corte del ruido en modo diferencial: 10 KHz
- Frecuencia de corte de ruido en modo común: 25 KHz

A la hora de diseñar una placa de circuito impreso para el filtro EMI, se deberá dejar un espacio mínimo de separación entre pistas por las que haya diferencias mayores a 200 VDC, de 2.5 mm entre ellas y un ancho de pista de 5 mm o más, con 35 µm de espesor, que deberán recubrirse de estaño.

## 2.2. Convertidor PFC

El convertidor PFC ha sido dimensionado teniendo en cuenta una red eléctrica de 230 VAC y 50 Hz, por lo que la tensión mínima que puede entrar al circuito es de 207 VAC. Todos los componentes se han dimensionado para dicha tensión mínima.

Se deberán utilizar un rectificador, transistor MOSFET y diodo Boost con capacidad para soportar 10A en conducción directa o más, y unas velocidades trabajo iguales o superiores; además las resistencias de conducción deberán ser iguales o inferiores, para así poder asegurar la compatibilidad con los cálculos de los disipadores. La bobina de almacenamiento, deberá tener una inductancia mínima de 560  $\mu$ H, así como ser capaz de soportar una corriente de 10A.

La salida del convertidor, deberá estar comprendida entre 380 y 420 VDC, por lo que el transistor MOSFET y el diodo Boost, deberán ser capaces de soportar tensiones, como mínimo, de 500 VDC y 600 VDC respectivamente. El puente de diodos del rectificador, soportará tensiones inversas de 600 VDC o más, así como una corriente nominal mínima en conducción directa de 10A.

Los condensadores de filtro en la salida, tendrán una tensión nominal de 450 VDC o más y una ESR de 0.29 ohmios o menos.

El driver para la conmutación del transistor MOSFET, deberá ser capaz de funcionar con una tensión de 12 VDC y ser capaz de proporcionar corrientes de pico de 1A.

Las resistencias que forman el divisor de tensión para la lectura analógica de la tensión de línea, así como la que forma el shunt para leer la corriente que circula por el circuito, deberán tener una precisión del 1% o mejor, así como una deriva térmica de 50 ppm/ $^{\circ}$ C o mejor.

A la hora de diseñar una placa de circuito impreso para el convertidor PFC, se deberá dejar un espacio mínimo de separación entre pistas por las que haya diferencias mayores a 200 VDC, de 2.5 mm entre ellas y un ancho de pista de 5 mm o más, con 35  $\mu$ m de espesor, que deberán recubrirse de estaño.

La potencia máxima que podrá suministrar el convertidor PFC es de 1000 W.

## 2.3. Convertidor DC – DC

El convertidor DC – DC deberá dimensionarse para resistir una tensión mínima a la entrada de 380 VDC y un máximo de 420 VDC, por lo que se deberá considerar utilizar transistores MOSFET en el puente que sean capaz de resistir como mínimo 500 VDC

La bobina de resonancia, deberá tener una inductancia mínima de 33  $\mu$ H y soportará corrientes de 10A o más.

El transformador para medir la corriente en el secundario tendrá una relación de transformación de 100:1 con el secundario.

Las resistencias que forman el shunt en el secundario, tendrán una potencia nominal de 5W y una precisión del 5%.

A la hora de diseñar una placa de circuito impreso para el convertidor DC – DC , se deberá dejar un espacio mínimo de separación entre pistas por las que haya diferencias mayores a 200 VDC, de 2.5 mm entre ellas y un ancho de pista de 5 mm o más en las pistas del primario y 7 mm o más en las del secundario, con 35  $\mu\text{m}$  de espesor, que deberán recubrirse de estaño.

Se deberá dejar 8 mm de separación mínima entre las pistas del primario y el secundario.

La potencia máxima a la que podrá suministrar el convertidor DC – DC es de 1000 W o 16A de corriente independientemente de la tensión de consigna.

## 2.4. Controlador de temperatura

Dado que la eficiencia es un requisito fundamental en este dispositivo, el control de temperatura y refrigeración debe interferir lo menos posible en ella, por ello, los MOSFET utilizados para el control de ventiladores deben tener una resistencia de encendido drenador – fuente de 28 miliohmios o menos para igualar las pérdidas originadas por la conducción.

Los ventiladores utilizados. Deben tener ser capaces de generar un flujo volumétrico de aire de 0.447  $\text{m}^3/\text{min}$  y consumir 0.25A o menos, adaptándose sus dimensiones y fijaciones al diseño mecánico.

Además, en esta placa, se llevan a cabo medidas analógicas de los termistores NTC, mediante el método del divisor de tensión. Por lo tanto, las resistencias que lo forman, deben tener una precisión nominal del 1% o mejor, así como una deriva térmica de 50 ppm/ $^{\circ}\text{C}$  o mejor.

Los disipadores utilizados en el convertidor PFC y el puente completo de transistores MOSFET en el convertidor DC – DC, tendrá una  $R_h$  de 4 $^{\circ}\text{C}/\text{W}$  o mejor. El disipador de los MOSFET de rectificación síncrona, tendrá una  $R_h$  de 5.3 $^{\circ}\text{C}/\text{W}$  o mejor.

## 2.5. Fuentes de alimentación auxiliares

El sistema de alimentación auxiliar, se encarga de proporcionar energía a los controladores de las diferentes etapas de potencia y a la gestión de la interfaz. Hay dos sistemas que deben permanecer separados galvánicamente en todo momento: el convertidor PFC y la electrónica del secundario del convertidor DC – DC. Para ello, se deben utilizar dos fuentes de alimentación diferentes, que deben cumplir las siguientes especificaciones:

- Fuente de alimentación controlador PFC: 12 VDC con capacidad para suministrar 0.5A o más.
- Fuente de alimentación del control en el secundario: 12 VDC con capacidad para suministrar 2A o más.

## 2.6. Controlador DC – DC

El controlador principal requiere de cierta precisión para realizar las medidas analógicas, es por ello, que se debe utilizar, en primer lugar, una referencia de tensión con una precisión obligatoriamente de 5V, con una precisión del 1% o mejor, así como una deriva térmica de 50 ppm/°C o mejor.

Las resistencias utilizadas en todos los circuitos de medida analógica o encargadas de configurar la ganancia de los amplificadores operacionales, deben tener una precisión del 1% o mejor, así como una deriva térmica de 50 ppm/°C o mejor.

## 2.7. Drivers de puente completo

Los drivers de control del puente completo, requieren de aislamiento galvánico entre la electrónica de control y los MOSFET, dado que se realiza un control desde el secundario. El aislamiento, será de 3000 VRMS entre el secundario y el primario o mejor, y un aislamiento entre puerta y puerta de 1500 VRMS o mejor.

Deben ser capaces de funcionar con una tensión de 12 VDC y proporcionar un pico de corriente de 4.5A o más. Además, deberán operar a una frecuencia de 100 KHz al 48% del ciclo de trabajo.

## 2.8. Drivers de rectificación síncrona

En caso de usar los drivers de rectificación síncrona controlada, los drivers implementados, no requerirán de aislamiento galvánico, al encontrarse en el lado secundario, igual que la electrónica que los controla.

Deben ser capaces de funcionar con una tensión de 12 VDC y proporcionar un pico de corriente de 4.5A o más. Además, deberán operar a una frecuencia de 100 KHz al 45% del ciclo de trabajo.

## 2.9. Conectores internos del dispositivo

Las conexiones internas entre la entrada de alterna, el filtro EMI, el convertidor PFC, el convertidor DC – DC y la etapa de salida se deben realizar con terminales Faston en formato de 6.3 mm con capacidad para resistir una corriente de 16A de forma continua.

## 2.10. Terminal de entrada CA e interruptor

El conector que suministra alimentación en corriente alterna proveniente de la red al sistema, debe ser un conector estándar de tres terminales C14 con capacidad para resistir 10A o más.

El interruptor, debe conmutar el estado de la fase y debe ser capaz de resistir una corriente de 10A.

## 2.11. Cableado interno

Para las conexiones internas de los dos convertidores principales del sistema, el filtro EMI y la etapa de salida, se conectarán con cable de cobre de 2.5 mm<sup>2</sup> de grosor.

El cableado que conecta las fuentes de alimentación auxiliares con la entrada AC y las diferentes partes del circuito que alimentan, deberán conectarse con cable de cobre de un grosor mínimo de 1.5 mm<sup>2</sup>.

Los ventiladores de la fuente, se conectarán con un cable de cobre de un grosor mínimo de 1 mm<sup>2</sup>.

El cableado de las comunicaciones entre las dos placas de control, se realizará con un cable de un grosor mínimo de 0.5 mm<sup>2</sup>.

## 2.12. Temperaturas máximas de operación

Los componentes utilizados, han sido dimensionados para una temperatura máxima de operación de 100°C. La temperatura máxima de funcionamiento a máxima potencia ha sido determinada como 90°C aproximadamente.

Dado que los cálculos se han realizado teniendo en cuenta una temperatura ambiente de 25°C, no se recomienda trabajar a máxima potencia con el dispositivo, en entornos por encima de los 35°C.

# 3. Parámetros de los protocolos de comunicaciones

Se detallan a continuación los parámetros utilizados para las comunicaciones UART e I2C utilizadas en el proyecto. Estos parámetros deberán respetarse para el correcto funcionamiento del sistema.

## 3.1. Parámetros UART

La comunicación serie UART se realiza con los siguientes parámetros de protocolo:

- Longitud de trama: 8 bits
- Paridad: Sin paridad
- Bit de Stop: 1
- Velocidad de transmisión: 38400 baudios

### 3.2. Parámetros I2C

La comunicación I2C se realiza con los siguientes parámetros de protocolo

- Velocidad: 100 KHz
- Dirección: 7 bits.

## 4. Parámetros recomendados de las piezas mecánicas

Este apartado refleja las condiciones de fabricación utilizadas en este proyecto y que se consideran recomendadas para conseguir una rigidez adecuada para la estructura.

### 4.1. Fabricación de piezas mediante impresión 3D

Para la fabricación de los paneles frontal y trasero, se ha utilizado impresión 3D en plástico PLA en una impresora BQ Hephestos 2 con los siguientes parámetros de impresión:

- Altura de capa: 0.2 mm
- Grosor de línea: 0.4 mm
- Ancho de la capa inferior/superior: 0.4 mm
- Relleno capa inicial: 100%
- Espesor de la pared: 1 mm
- Número de perímetros: 2
- Espesor capa inferior: 1 mm
- Número de capas inferiores: 5
- Espesor capa superior: 1 mm
- Capas de capas superiores: 5
- Densidad de relleno: 20%
- Tipo de relleno: Grid
- Temperatura de impresión: 210°C
- Habilitar retracción
- Distancia de retracción: 6.5 mm
- Velocidad de retracción: 25 mm/s
- Velocidad de impresión: 50 mm/s
- Habilitar refrigeración
- Refrigeración: 100%
- Soportes: Sí

## 4.2. Propiedades de la carcasa

El material elegido debe ser una lámina de acero en bruto de 1 mm de espesor, o un material cuya resistencia mecánica a la flexión sea igual o mejor con el mismo espesor.

## 5. Software

Con el objeto de hacer uso del software de control remoto de la fuente de alimentación, se debe disponer de un ordenador con el sistema operativo Windows 10 en su versión de 64 bits y el motor Run-Time de LabVIEW de la versión 15.

Para acceder a la ejecución y modificación del código fuente suministrado, así como la modificación y reproducción de las piezas mecánicas, se debe disponer de los siguientes programas en las versiones que aquí se describen o compatibles:

- Arduino IDE v1.8.9
- MPLAB X IDE v5.10
- MPLAB CODE CONFIGURATOR v3.85
- LabVIEW 2015
- Nextion Editor v0.58
- MATLAB 2016a
- Eagle CAD v9.2.2
- SATURN PCB Toolkit v7.08
- Fusion 360 versión del 22 de agosto 2019
- Cura 3D v3.4.1

Firmado, a 31 de agosto de 2019, Logroño



Alberto Otero Olavarrieta



# MEDICIONES



## Índice

1.	Materiales y mano de obra para la construcción del prototipo .....	351
1.1.	Controlador PFC.....	351
1.2.	Entrada y Filtro EMI.....	352
1.3.	Convertidor Boost PFC.....	353
1.4.	Convertidor DC-DC Puente Completo ZVS .....	354
1.5.	Driver Puente Completo .....	355
1.6.	Driver Rectificación Síncrona .....	355
1.7.	Controlador Convertidor Puente Completo .....	356
1.8.	Controlador Panel Frontal .....	357
1.9.	Botones y USB.....	358
1.10.	Control de temperatura.....	359
1.11.	Etapa de Salida.....	359
1.12.	Piezas Auxiliares y pequeño material .....	360
2.	Mano de obra de desarrollo.....	361
2.1.	Controlador PFC.....	361
2.2.	Entrada y filtro EMI.....	361
2.3.	Convertidor Boost PFC.....	361
2.4.	Convertidor DC – DC ZVS .....	361
2.5.	Driver Puente Completo .....	361
2.6.	Driver Rectificación Síncrona .....	361
2.7.	Controlador convertidor puente completo.....	362
2.8.	Controlador panel frontal.....	362
2.9.	Botones y USB.....	362
2.10.	Control de temperatura.....	362
2.11.	Etapa de salida .....	362
2.12.	Piezas auxiliares.....	362



## 1. Materiales y mano de obra para la construcción del prototipo

A continuación, se considerará un capítulo/apartado por cada uno de los bloques fundamentales que componen el equipo, de modo que, si se debe sustituir un elemento dañado, este pueda ser fácilmente medido y presupuestado.

### 1.1. Controlador PFC

Parte	Valor	Encapsulado/medidas	Cantidad
Conector 400VDC, VCC	JST Header Hembra 2.54 mm 2A	KK-156-2	2
C1	1nF 25V	C5B4.5	1
C2	100uF 25V	E2,5-6	1
C3	4.7nF 25V	C5B4.5	1
C4	100pF 25V	C5B4.5	1
C5	10nF 25V	C5B4.5	1
C6	10pF 25V	C5B4.5	1
IC1	ICE3PCS01	SO14	1
JP1	Pin Header Macho 2.54 mm 3A	1X06	1
PCB	Dos capas - 35u Cu, 1.6mm FR-4	49x59 mm	1
R1	220R 1/4W	0207/7	1
R10	62k 1% 1/4W	0207/7	1
R11, R12	2M 1% 1/4W	0207/7	2
R13	24k 1% 1/4W	0207/7	1
R14	560k 1% 1/4W	0207/7	1
R2	43k 1% 1/4W	0207/7	1
R3	36k 1/4W	0207/7	1
R4	10k 1/4W	0207/7	1
R5	330k 1/4W	0207/7	1
R6	200k 1/4W	0207/7	1
R7, R8	1.5M 1% 1/4W	0207/7	2
R9	27k 1% 1/4W	0207/7	1
Horas de montaje	Hora de montador	Horas / hombre	5
Horas de ajuste	Hora de montador	Horas / hombre	5

## 1.2. Entrada y Filtro EMI

Parte	Valor	Encapsulado/medidas	Cantidad
C1, C2	1.2uF 275VAC	C325-137X374	2
C3, C4	4n7 3KV	C5B4.5	2
Terminales Faston EARTH, L, N, X1, X2, X3, X4, X5, X6	Faston Terminal macho 6.3mm 16A	W237-102	9
EARTHGND, FASE, NEUTRO	Pin Header Macho 3A/pin	2x05	3
K1	SLA-12VDC	SLA-12VDC	1
L1, L2	100uH	SFT1030	2
L3	5mH		1
R1	1M 1/4W	0309/12	1
R2, R3	100R 5W	0922/22	2
R4, R5, R6	S20K420	S20K420	3
CTRLRELE	JST Header Hembra 2.54 mm 2A	1X02/90	1
F1	10A FUSE HRC 38mm	38MMFUSE	1
F2	10A FUSE HRC 38mm	38MMFUSE	1
PCB	Dos capas - 35u Cu, 1.6mm FR-4	197x52 mm	1
Horas de montaje	Hora de montador	Horas / hombre	10
Horas de ajuste	Hora de montador	Horas / hombre	5

### 1.3. Convertidor Boost PFC

Parte	Valor	Encapsulado/ medidas	Cantidad
Terminales Faston 400VDC, 400VDC1, GND, GND2	Faston Terminal macho 6.3mm 16A	W237-102	4
B1	KBU1510	2KBB-R	1
C1	4n7 3KV	C5B5	1
C11	100nF 25V	C075-032X103	1
C5	2.2uF 630V	C325-137X374	1
C6	10nF 25V	C5B4.5	1
C7, C8	470uF 450V	EB35D	2
C9, C10	100nF 25V	C5B4.5	2
D1, D3, D5	1N4148DO35-10	DO35-10	3
D2	1N5400	DO201-15	1
D4	MUR3060PT	TO 247	1
Conectores EARTHGND, L, N	Pin Header hembra 3A/pin	2x05	3
IC1	IR2125	DIL08	1
L4	2 mH	SFT1240	1
PCB	Dos capas - 35u Cu, 1.6mm FR-4	120x240 mm	1
Q1	IRFP460PBF	TO 247	1
R4	10k 1/4W	0309/12	1
R5	100R 1/4W	0309/12	1
R6	0.01R 5W	0922/22	1
R7	10R 1/2W	0309/12	1
R8, R9, R10, R11	100k 1W	0411/15	4
CTRLRELE	Pin Header Macho 2.54 mm 3A	1X02	1
Conector SV1	Pin Header Macho 2.54 mm 3A	1x06	1
T1	BC547	TO92	1
HT1	Heatsink 40x40x80	40x40x80 mm	1
Horas de montaje	Hora de montador	Horas / hombre	10
Horas de ajuste	Hora de montador	Horas / hombre	10

## 1.4. Convertidor DC-DC Puente Completo ZVS

Parte	Valor	Encapsulado/ medidas	Cantidad
Terminales Faston 400VDC, 400VDC-2, GNDOUT, GNDOUT2, INPUTGND, INPUTGND-2, SEC2, SEC3, SEC5, SEC6, VOUT, VOUT2	Faston Terminal macho 6.3mm 16A	W237-102	12
C1, C2, C3	470uF 100V	E7,5-16	3
C4	1000pF 3KV	C7.5B5	1
C5, C6, C7	1uF 100V	C150-084X183	3
C8	470nF 630V	C150-084X183	1
C9	200nF 250V	C10B6	1
D1, D2	MUR860	TO220ACS	2
D3	1n4148	DO41-7.6	1
D3	MUR3060PT	TO247BV	1
Tira de pines DRIVER1, DRIVER2	Pin Header Hembra 2.54 mm 3A	1X04	2
Tira de pines JP1, JP2, JP3, JP4, JP5	Pin Header Hembra 2.54 mm 3A/Pin	2X05	5
KK1	SK104	SK104	1
L1, L2	33uH	SFT1240	2
PCB	Dos capas - 35u Cu, 1.6mm FR-4	125x265 mm	1
Q1, Q2, Q3, Q4	IRFP460	TO247BV	4
Q5, Q6	IRFP90N20D	TO247BV	2
R11, R12	0.02R 5W	0309/10	2
R3, R4	1k 1W	0411/15	2
R5	100k 1W	0411/15	1
R6	4k7 1% 1/4W	0207/10	1
R7, R8	180R 1% 1/4W	0207/10	2
SYNCRDRIVER	Pin Header	1X03	1
TF1	PE-63587NL	PE-63587NL	1
TF2	EE55	EE55	1
HT1	Heatsink 40x40x80	40x40x80	1
HT2	CR101-50	CR101-51	1
Horas de montaje	Hora de montador	Horas / hombre	10
Horas de ajuste	Hora de montador	Horas / hombre	10

## 1.5. Driver Puente Completo

Parte	Valor	Encapsulado/medidas	Cantidad
C1	1uF 25V	C7.5B5	1
D1, D2	1N4148DO35-10	DO35-10	2
IC1	MCP1404	DIL08	1
INPUT	JST Header Hembra 2.54 mm 2A	1x02	1
OUT	Pin Header Macho 2.54 mm 3A	1X04/90	1
R1, R3	22R 1/2W	0207/10	2
R2, R4	4k7 1/4W	0207/10	2
R5	10R 1W	0207/10	1
TF1	P0584NL	P0584NL	1
PCB	Dos capas - 35u Cu, 1.6mm FR-4	31x47 mm	1
Horas de montaje	Hora de montador	Horas / hombre	5
Horas de ajuste	Hora de montador	Horas / hombre	5

## 1.6. Driver Rectificación Síncrona

Parte	Valor	Encapsulado/medidas	Cantidad
C1	1uF	C7.5B4	1
D1, D2	1N4148DO35-10	DO35-10	2
IC1	MCP1404	DIL08	1
INPUT	JST Header Hembra 2.54 mm 2A	1x02	1
OUT	Pin Header Macho 2.54 mm 3A	1X03/90	1
PCB	Dos capas - 35u Cu, 1.6mm FR-4	23x37 mm	1
R1, R2	3R3 1/2W	0207/10	2
R3, R4	47K 1/4W	0207/10	2
Horas de montaje	Hora de montador	Horas / hombre	5
Horas de ajuste	Hora de montador	Horas / hombre	5

## 1.7. Controlador Convertidor Puente Completo

Parte	Valor	Encapsulado/medidas	Cantidad
12V, BITTEMP, I2C, RELEOUT, VOUT	JST Header Hembra 2.54 mm 2A	1x02	5
C1, C2, C9	100nF 25V	C5B3.5	3
C10	30pF 25V	C5B3.5	1
C11	30pF 25v	C5B3.5	1
C14	1uF 25V	C5B3.5	1
C3, C4, C8, C13	10uF 25V	E2,5-6E	4
C5, C6, C12, C15, C16	100nF 25V	C5B2.5	5
IC1	ICL7660CSA	SO08	1
IC2	INA126P	DIL08	1
IC3	REG1117	SOT223	1
IC5	MCP6V26	SO08	1
Tira de pines JP1, JP2, JP3	Pin Header	2X05	3
PCB	Dos capas - 35u Cu, 1.6mm FR-4	58x93 mm	1
PWM1, PWM2, PWM3	JST Header Hembra 2.54 mm 2A	1x04	3
Q1	8Mhz	HC49U-V	1
R1	10k 1% 1/4W	0207/7	1
R11	4k7 1% 1/4W	0207/7	1
R14	1k 1% 1/4W	0207/7	1
R15	220R 1/4W	0207/7	1
R16	10k 1/4W	0207/7	1
R2, R3, R19	2k2 1/4W	0207/7	3
R20	1k 1/4W	0207/7	1
R4	120k 1% 1/4W	0207/7	1
R5, R8	10k 1% 1/4W	0207/7	2
R9	330R 1/4W	0207/7	1
SV1	Pin Header	FE06	1
T3	2N2222	TO92-EBC	1
UIC4	DSPIC33EV256GM002	DIL28-3	1
VR1	TL431A	TO92-CLP	1
Horas de montaje	Hora de montador	Horas / hombre	10
Horas de ajuste	Hora de montador	Horas / hombre	10

## 1.8. Controlador Panel Frontal

Parte	Valor	Encapsulado/medidas	Cantidad
5V, I2C, PANTALLA, VCC	JST Header Hembra 2.54 mm 2A	1x02	4
C1, C2, C7, C8	22pF 25V	C5B2.5	4
C3, C4	100nF 25V	C5B2.5	2
C5, C6	100nF 25V	C5B4.5	2
D1	1N4148DO35-7	DO35-7	1
IC1	7805TV	TO220V	1
IC2	ATMEGA328-20P	DIL28-3	1
IC3	CH340G	SOIC16	1
JP1	Pin Header Macho 2.54 mm 3A	1X08	1
JP2	Pin Header Macho 2.54 mm 3A	1X07	1
JP3	Pin Header Macho 2.54 mm 3A	1X04	1
PCB	Dos capas - 35u Cu, 1.6mm FR-4	106x46 mm	1
Q1	12MHz	HC49/S	1
Q2	16MHz	HC49/S	1
R1, R2	1k 1/4W	0207/7	2
Horas de montaje	Hora de montador	Horas / hombre	5
Horas de ajuste	Hora de montador	Horas / hombre	5

## 1.9. Botones y USB

Parte	Valor	Encapsulado/medidas	Cantidad
C1, C2, C3, C4	1uF 25V	E2-5	4
C5, C6, C7, C8	100nF 25V	C5B3	4
IC1, IC3	4013D	SO14	2
IC2, IC4	7414D	SO14	2
Tira de pines JP1	Pin Header Hembra 2.54 mm 3A	1X08	1
Tira de pines JP2	Pin Header Hembra 2.54 mm 3A	1X07	1
Tira de pines JP3	Pin Header Hembra 2.54 mm 3A	1X04	1
PCB	Dos capas - 35u Cu, 1.6mm FR-4	148x47 mm	1
R1, R2, R3, R4, R5, R6, R7, R8, R9, R10	10k 1/4W	0207/10	10
R11, R12	1K 1/4W	0207/10	2
R15	1k 1/4W	0207/10	1
S1, S2	Encoder	B3F-10XX	2
SG2	F/CM12P	F/CM12P	1
Encoder SW1, SW4	EC12E_SW	ALPS_EC12E_SW	2
T1	BC547	TO92-EBC	1
X1	KUSBVX-BS1N	KUSBVX-BS1N	1
Horas de montaje	Hora de montador	Horas / hombre	5
Horas de ajuste	Hora de montador	Horas / hombre	5

## 1.10. Control de temperatura

Parte	Valor	Encapsulado/medidas	Cantidad
C1, C2	100nF 25V	C5B5	2
D1, D2	1N4004	DO41-10	2
Ventilador FAN1, FAN2, FAN3, FAN4, NTCPUENTE, NTCRECT, OVERTEMP, VCC	JST Header Hembra 2.54 mm 2A	1x02	8
IC1	TINY84-PU	DIL14	1
IC2	78L05Z	TO92	1
KK1, KK2	SK95-2M3	SK95-2M3	2
PCB	Dos capas - 35u Cu, 1.6mm FR-4	57x60 mm	1
Q1, Q2	IRFZ44	TO220BV	2
R1, R3	1k 1/4W	0207/10	2
R2, R4	10k 1/4W	0207/10	2
R5	10k 1% 1/4W	0207/10	1
R6	10k 1% 1/4W	0207/10	1
SV1, SV2	Pin Header Macho 2.54 mm 3A	2x05	2
Horas de montaje	Hora de montador	Horas / hombre	5
Horas de ajuste	Hora de montador	Horas / hombre	5

## 1.11. Etapa de Salida

Parte	Valor	Encapsulado/medidas	Cantidad
Terminal Faston GND1, GND2, GND3, VOUT1, VOUT2, VOUT3	Faston Terminal macho 6.3mm 16A	W237-102	6
CTRLRELE, VOUT	JST Header Hembra 2.54 mm 2A	1x02	2
K1	SLA-12VDC	SLA-12VDC	1
PCB	Dos capas - 35u Cu, 1.6mm FR-4	31x122 mm	1
Horas de montaje	Hora de montador	Horas / hombre	5
Horas de ajuste	Hora de montador	Horas / hombre	5

## 1.12. Piezas Auxiliares y pequeño material

Parte	Valor/medidas	Cantidad
Panel Frontal PLA	Kg PLA	0.116
Panel Trasero PLA	Kg PLA	0.109
Fuente de alimentación 12V1A	12V 1A	1
Fuente de alimentación 12V2A	12V 2A	1
Conector AC 3 terminales	C14	1
Terminal 4 mm	4 mm 30A	2
Interruptor AC 10A		1
Cubrebotón Encoders		2
Pantalla Nextion 3.2"		1
Ventiladores	Delta FFB0412SVHN	4
Terminales Faston Hembra	Faston Hembra 6.3 mm	25
Cable 2.5 mm (m)	30 mm	6
Cable 1.5 mm (m)	30 mm	6
PicKit 3		1
Tapa de acero inferior	295 x 280 mm	1
Tapa de acero superior	446 x 280 mm	1
Tornillos M3 5 mm	M3x5	11
Tornillos M3 10 mm	M3x10	11
Tuercas M3	M3	11
Separadores M3 5 mm	M3x5	11
Tirafondos M3 8 mm	M3x8	8
NTC B3950	10k	2
Horas de montaje	Horas / hombre	10
Horas de ajuste	Horas / hombre	10

## 2. Mano de obra de desarrollo

### 2.1. Controlador PFC

Tipo de trabajo	Cantidad
Horas de diseño	20h
Horas de documentación	5h

### 2.2. Entrada y filtro EMI

Tipo de trabajo	Cantidad
Horas de diseño	30h
Horas de documentación	5h

### 2.3. Convertidor Boost PFC

Tipo de trabajo	Cantidad
Horas de diseño	30h
Horas de documentación	15h

### 2.4. Convertidor DC – DC ZVS

Tipo de trabajo	Cantidad
Horas de diseño	40h
Horas de documentación	15h

### 2.5. Driver Puente Completo

Tipo de trabajo	Cantidad
Horas de diseño	10h
Horas de documentación	5h

### 2.6. Driver Rectificación Síncrona

Tipo de trabajo	Cantidad
Horas de diseño	10h
Horas de documentación	5h

## 2.7. Controlador convertidor puente completo

Tipo de trabajo	Cantidad
Horas de diseño	50h
Horas de programación	100h
Horas de depuración	20h
Horas de documentación	15h

## 2.8. Controlador panel frontal

Tipo de trabajo	Cantidad
Horas de diseño	25h
Horas de programación	100h
Horas de depuración	10h
Horas de documentación	15h

## 2.9. Botones y USB

Tipo de trabajo	Cantidad
Horas de diseño	10h
Horas de documentación	5h

## 2.10. Control de temperatura

Tipo de trabajo	Cantidad
Horas de diseño	20h
Horas de programación	30h
Horas de depuración	10h
Horas de documentación	10h

## 2.11. Etapa de salida

Tipo de trabajo	Cantidad
Horas de diseño	5h
Horas de documentación	5h

## 2.12. Piezas auxiliares

Tipo de trabajo	Cantidad
Horas de diseño	50h
Horas de documentación	10h

# PRESUPUESTO



## Índice

1.	Listado de precios unitarios .....	366
1.1.	Materiales y mano de obra para la construcción del prototipo .....	366
2.	Listado de precios descompuestos .....	367
2.1.	Controlador PFC.....	367
2.2.	Entrada y Filtro EMI.....	368
2.3.	Convertidor Boost PFC.....	369
2.4.	Convertidor DC-DC Puente Completo ZVS .....	370
2.5.	Driver Puente Completo .....	371
2.6.	Driver Rectificación Síncrona .....	371
2.7.	Controlador Convertidor Puente Completo .....	372
2.8.	Controlador Panel Frontal .....	373
2.9.	Botones y USB.....	374
2.10.	Control de temperatura.....	375
2.11.	Etapas de Salida.....	375
2.12.	Piezas Auxiliares y pequeño material .....	376
2.13.	Subtotal Materiales y elementos para la construcción del prototipo montados y ajustados .....	377
2.14.	Subtotal Mano de obra de desarrollo .....	377
3.	Total costes directos .....	378
4.	Coste general.....	378
5.	Total libre de impuestos .....	378
6.	Total final .....	378

## 1. Listado de precios unitarios

### 1.1. Materiales y mano de obra para la construcción del prototipo

Módulo	Precio
PCB Controlador PFC	210,92 €
PCB Entrada AC y Filtro EMI	325,13 €
PCB Convertidor PFC	446,70 €
PCB Convertidor DC-DC	466,58 €
PCB Control del convertidor	416,02 €
Driver FB x2	426,62 €
Driver SR	209,34 €
PCB Control de temperatura	211,98 €
PCB Control Panel Frontal	214,15 €
PCB Botones y USB	216,22 €
Etapas de Salida	212,32 €
Piezas auxiliares y pequeño material	508,07 €

A continuación, se relacionan las horas empleadas en el diseño e implementación completa del primer prototipo. Para unidades sucesivas, sólo serían necesarias las horas relativas a montaje y ajuste.

Tipo de trabajo	Cantidad	Precio
Horas de diseño	1h	30€
Horas de programación	1h	30€
Horas de depuración	1h	30€
Horas de montaje	1h	20€
Horas de ajuste	1h	20€
Horas de documentación	1h	20€

## 2. Listado de precios descompuestos

Los precios aquí presentados, se han obtenido fruto de la compra individual de los componentes, o en lotes muy pequeños. No se aplican descuentos por volumen.

### 2.1. Controlador PFC

Parte	Valor	Encapsulado/ medidas	Cantidad	Precio ud	Precio
Conector 400VDC, VCC	JST Header Hembra 2.54 mm 2A	KK-156-2	2	0,09 €	0,18 €
C1	1nF 25V	C5B4.5	1	0,08 €	0,08 €
C2	100uF 25V	E2,5-6	1	0,12 €	0,12 €
C3	4.7nF 25V	C5B4.5	1	0,06 €	0,06 €
C4	100pF 25V	C5B4.5	1	0,05 €	0,05 €
C5	10nF 25V	C5B4.5	1	0,06 €	0,06 €
C6	10pF 25V	C5B4.5	1	0,05 €	0,05 €
IC1	ICE3PCS01	SO14	1	2,11 €	2,11 €
JP1	Pin Header Macho 2.54 mm 3A	1X06	1	0,05 €	0,05 €
PCB	Dos capas - 35u Cu, 1.6mm FR-4	49x59 mm	1	7,51 €	7,51 €
R1	220R 1/4W	0207/7	1	0,04 €	0,04 €
R10	62k 1% 1/4W	0207/7	1	0,05 €	0,05 €
R11, R12	2M 1% 1/4W	0207/7	2	0,05 €	0,10 €
R13	24k 1% 1/4W	0207/7	1	0,05 €	0,05 €
R14	560k 1% 1/4W	0207/7	1	0,05 €	0,05 €
R2	43k 1% 1/4W	0207/7	1	0,05 €	0,05 €
R3	36k 1/4W	0207/7	1	0,04 €	0,04 €
R4	10k 1/4W	0207/7	1	0,04 €	0,04 €
R5	330k 1/4W	0207/7	1	0,04 €	0,04 €
R6	200k 1/4W	0207/7	1	0,04 €	0,04 €
R7, R8	1.5M 1% 1/4W	0207/7	2	0,05 €	0,10 €
R9	27k 1% 1/4W	0207/7	1	0,05 €	0,05 €
Horas de montaje	Hora de montador	Horas / hombre	5	20€	100€
Horas de ajuste	Hora de montador	Horas / hombre	5	20€	100€

## 2.2. Entrada y Filtro EMI

Parte	Valor	Encapsulado/ medidas	Cantidad	Precio ud	Precio
C1, C2	1.2uF 275VAC	C325-137X374	2	0,78 €	1,56 €
C3, C4	4n7 3KV	C5B4.5	2	0,08 €	0,16 €
Terminales Faston EARTH, L, N, X1, X2, X3, X4, X5, X6	Faston Terminal macho 6.3mm 16A	W237-102	9	0,11 €	0,99 €
EARTHGND, FASE, NEUTRO	Pin Header Macho 3A/pin	2x05	3	0,12 €	0,36 €
K1	SLA-12VDC	SLA-12VDC	1	0,85 €	0,85 €
L1, L2	100uH	SFT1030	1	0,85 €	0,85 €
L3	5mH		1	0,95 €	0,95 €
R1	1M 1/4W	0309/12	2	0,64 €	1,28 €
R2, R3	100R 5W	0922/22	1	2,68 €	2,68 €
R4, R5, R6	S20K420	S20K420	1	13,45 €	13,45 €
CTRLRELE	JST Header Hembra 2.54 mm 2A	1X02/90	1	0,04 €	0,04 €
F1	10A FUSE HRC 38mm	38MMFUSE	2	0,23 €	0,46 €
F2	10A FUSE HRC 38mm	38MMFUSE	3	0,47 €	1,41 €
PCB	Dos capas - 35u Cu, 1.6mm FR-4	197x52 mm	1	0,09 €	0,09 €
Horas de montaje	Hora de montador	Horas / hombre	10	20€	200€
Horas de ajuste	Hora de montador	Horas / hombre	5	20€	100€

### 2.3. Convertidor Boost PFC

Parte	Valor	Encapsulado/ medidas	Cantidad	Precio ud	Precio
Terminales Faston 400VDC, 400VDC1, GND, GND2	Faston Terminal macho 6.3mm 16A	W237-102	4	0,11 €	0,44 €
B1	KBU1510	2KBB-R	1	0,95 €	0,95 €
C1	4n7 3KV	C5B5	1	0,09 €	0,09 €
C11	100nF 25V	C075-032X103	1	0,08 €	0,08 €
C5	2.2uF 630V	C325-137X374	1	0,58 €	0,58 €
C6	10nF 25V	C5B4.5	1	0,08 €	0,08 €
C7, C8	470uF 450V	EB35D	2	0,22 €	0,44 €
C9, C10	100nF 25V	C5B4.5	2	0,08 €	0,16 €
D1, D3, D5	1N4148DO35-10	DO35-10	3	0,08 €	0,24 €
D2	1N5400	DO201-15	1	0,12 €	0,12 €
D4	MUR3060PT	TO247	1	0,68 €	0,68 €
Conectores EARTHGND, L, N	Pin Header hembra 3A/pin	2x05	3	0,14 €	0,42 €
IC1	IR2125	DIL08	1	5,64 €	5,64 €
L4	2 mH	SFT1240	1	0,23 €	0,23 €
PCB	Dos capas - 35u Cu, 1.6mm FR-4	120x240 mm	1	11,96 €	11,96 €
Q1	IRFP460PBF	TO247	1	22,61 €	22,61 €
R4	10k 1/4W	0309/12	1	1,08 €	1,08 €
R5	100R 1/4W	0309/12	1	0,04 €	0,04 €
R6	0.01R 5W	0922/22	1	0,04 €	0,04 €
R7	10R 1/2W	0309/12	1	0,21 €	0,21 €
R8, R9, R10, R11	100k 1W	0411/15	1	0,06 €	0,06 €
CTRLRELE	Pin Header Macho 2.54 mm 3A	1X02	4	0,08 €	0,32 €
Conector SV1	Pin Header Macho 2.54 mm 3A	1x06	1	0,04 €	0,04 €
T1	BC547	TO92	1	0,05 €	0,05 €
HT1	Heatsink 40x40x80	40x40x80 mm	1	0,14 €	0,14 €
Horas de montaje	Hora de montador	Horas / hombre	10	20€	200€
Horas de ajuste	Hora de montador	Horas / hombre	10	20€	200€

## 2.4. Convertidor DC-DC Puente Completo ZVS

Parte	Valor	Encapsulado/ medidas	Cantidad	Precio ud	Precio
Terminales Faston 400VDC, 400VDC-2, GNDOUT, GNDOUT2, INPUTGND, INPUTGND-2, SEC2, SEC3, SEC5, SEC6, VOUT, VOUT2	Faston Terminal macho 6.3mm 16A	W237-102	12	0,11 €	1,32 €
C1, C2, C3	470uF 100V	E7,5-16	3	0,21 €	0,63 €
C4	1000pF 3KV	C7.5B5	1	0,14 €	0,14 €
C5, C6, C7	1uF 100V	C150-084X183	3	0,09 €	0,27 €
C8	470nF 630V	C150-084X183	1	0,25 €	0,25 €
C9	200nF 250V	C10B6	1	0,12 €	0,12 €
D1, D2	MUR860	TO220ACS	2	0,26 €	0,52 €
D3	1n4148	DO41-7.6	1	0,08 €	0,08 €
D3	MUR3060PT	TO247BV	1	0,68 €	0,68 €
Tira de pines DRIVER1, DRIVER2	Pin Header Hembra 2.54 mm 3A	1X04	2	0,06 €	0,12 €
Tira de pines JP1, JP2, JP3, JP4, JP5	Pin Header Hembra 2.54 mm 3A/Pin	2X05	1	5,64 €	5,64 €
KK1	SK104	SK104	1	5,25 €	5,25 €
L1, L2	33uH	SFT1240	5	0,08 €	0,40 €
PCB	Dos capas - 35u Cu, 1.6mm FR-4	125x265 mm	1	0,46 €	0,46 €
Q1, Q2, Q3, Q4	IRFP460	TO247BV	2	2,57 €	5,14 €
Q5, Q6	IRFP90N20D	TO247BV	1	24,82 €	24,82 €
R11, R12	0.02R 5W	0309/10	4	1,08 €	4,32 €
R3, R4	1k 1W	0411/15	2	2,47 €	4,94 €
R5	100k 1W	0411/15	2	0,26 €	0,52 €
R6	4k7 1% 1/4W	0207/10	2	0,08 €	0,16 €
R7, R8	180R 1% 1/4W	0207/10	1	0,08 €	0,08 €
SYNCRDRIVER	Pin Header	1X03	1	0,05 €	0,05 €
TF1	PE-63587NL	PE-63587NL	2	0,05 €	0,10 €
TF2	EE55	EE55	1	0,04 €	0,04 €
HT1	Heatsink 40x40x80	40x40x80	1	4,58 €	4,58 €
HT2	CR101-50	CR101-51	1	5,95 €	5,95 €
Horas de montaje	Hora de montador	Horas / hombre	10	20€	200€
Horas de ajuste	Hora de montador	Horas / hombre	10	20€	200€

## 2.5. Driver Puente Completo

Parte	Valor	Encapsulado/ medidas	Cantidad	Precio ud	Precio
C1	1uF 25V	C7.5B5	1	0,08 €	0,08 €
D1, D2	1N4148DO35-10	DO35-10	2	0,08 €	0,16 €
IC1	MCP1404	DIL08	1	1,20 €	1,20 €
INPUT	JST Header Hembra 2.54 mm 2A	1x02	1	0,11 €	0,11 €
OUT	Pin Header Macho 2.54 mm 3A	1X04/90	1	0,05 €	0,05 €
R1, R3	22R 1/2W	0207/10	1	7,54 €	7,54 €
R2, R4	4k7 1/4W	0207/10	2	0,06 €	0,12 €
R5	10R 1W	0207/10	2	0,04 €	0,08 €
TF1	P0584NL	P0584NL	1	0,08 €	0,08 €
PCB	Dos capas - 35u Cu, 1.6mm FR-4	31x47 mm	1	3,97 €	3,97 €
Horas de montaje	Hora de montador	Horas / hombre	5	20€	100€
Horas de ajuste	Hora de montador	Horas / hombre	5	20€	100€

## 2.6. Driver Rectificación Síncrona

Parte	Valor	Encapsulado/ medidas	Cantidad	Precio ud	Precio
C1	1uF	C7.5B4	1	0,08 €	0,08 €
D1, D2	1N4148DO35-10	DO35-10	2	0,08 €	0,16 €
IC1	MCP1404	DIL08	1	1,20 €	1,20 €
INPUT	JST Header Hembra 2.54 mm 2A	1x02	1	0,11 €	0,11 €
OUT	Pin Header Macho 2.54 mm 3A	1X03/90	1	0,05 €	0,05 €
PCB	Dos capas - 35u Cu, 1.6mm FR-4	23x37 mm	1	7,54 €	7,54 €
R1, R2	3R3 1/2W	0207/10	2	0,06 €	0,12 €
R3, R4	47K 1/4W	0207/10	2	0,04 €	0,08 €
Horas de montaje	Hora de montador	Horas / hombre	5	20€	100€
Horas de ajuste	Hora de montador	Horas / hombre	5	20€	100€

## 2.7. Controlador Convertidor Puente Completo

Parte	Valor	Encapsulado/ medidas	Cantidad	Precio ud	Precio
12V, BITTEMP, I2C, RELEOUT, VOUT	JST Header Hembra 2.54 mm 2A	1x02	5	0,09 €	0,45 €
C1, C2, C9	100nF 25V	C5B3.5	3	0,08 €	0,24 €
C10	30pF 25V	C5B3.5	1	0,06 €	0,06 €
C11	30pF 25v	C5B3.5	1	0,06 €	0,06 €
C14	1uF 25V	C5B3.5	1	0,08 €	0,08 €
C3, C4, C8, C13	10uF 25V	E2,5-6E	4	0,11 €	0,44 €
C5, C6, C12, C15, C16	100nF 25V	C5B2.5	5	0,08 €	0,40 €
IC1	ICL7660CSA	SO08	1	0,41 €	0,41 €
IC2	INA126P	DIL08	1	1,15 €	1,15 €
IC3	REG1117	SOT223	1	0,14 €	0,14 €
IC5	MCP6V26	SO08	1	1,04 €	1,04 €
Tira de pines JP1, JP2, JP3	Pin Header	2X05	3	0,05 €	0,15 €
PCB	Dos capas - 35u Cu, 1.6mm FR-4	58x93 mm	1	7,62 €	7,62 €
PWM1, PWM2, PWM3	JST Header Hembra 2.54 mm 2A	1x04	3	0,05 €	0,15 €
Q1	8Mhz	HC49U-V	1	0,08 €	0,08 €
R1	10k 1% 1/4W	0207/7	1	0,05 €	0,05 €
R11	4k7 1% 1/4W	0207/7	1	0,05 €	0,05 €
R14	1k 1% 1/4W	0207/7	1	0,05 €	0,05 €
R15	220R 1/4W	0207/7	1	0,04 €	0,04 €
R16	10k 1/4W	0207/7	1	0,04 €	0,04 €
R2, R3, R19	2k2 1/4W	0207/7	3	0,04 €	0,12 €
R20	1k 1/4W	0207/7	1	0,04 €	0,04 €
R4	120k 1% 1/4W	0207/7	1	0,05 €	0,05 €
R5, R8	10k 1% 1/4W	0207/7	2	0,05 €	0,10 €
R9	330R 1/4W	0207/7	1	0,04 €	0,04 €
SV1	Pin Header	FE06	1	0,07 €	0,07 €
T3	2N2222	TO92-EBC	1	0,12 €	0,12 €
UIC4	DSPIC33EV256GM002	DIL28-3	1	2,46 €	2,46 €
VR1	TL431A	TO92-CLP	1	0,32 €	0,32 €
Horas de montaje	Hora de montador	Horas / hombre	10	20€	200€
Horas de ajuste	Hora de montador	Horas / hombre	10	20€	200€

## 2.8. Controlador Panel Frontal

Parte	Valor	Encapsulado/ medidas	Cantidad	Precio ud	Precio
5V, I2C, PANTALLA, VCC	JST Header Hembra 2.54 mm 2A	1x02	4	0,08 €	0,32 €
C1, C2, C7, C8	22pF 25V	C5B2.5	4	0,06 €	0,24 €
C3, C4	100nF 25V	C5B2.5	2	0,08 €	0,16 €
C5, C6	100nF 25V	C5B4.5	2	0,08 €	0,16 €
D1	1N4148DO35-7	DO35-7	1	0,08 €	0,08 €
IC1	7805TV	TO220V	1	0,18 €	0,18 €
IC2	ATMEGA328-20P	DIL28-3	1	1,34 €	1,34 €
IC3	CH340G	SOIC16	1	0,34 €	0,34 €
JP1	Pin Header Macho 2.54 mm 3A	1X08	1	0,08 €	0,08 €
JP2	Pin Header Macho 2.54 mm 3A	1X07	1	0,08 €	0,08 €
JP3	Pin Header Macho 2.54 mm 3A	1X04	1	0,06 €	0,06 €
PCB	Dos capas - 35u Cu, 1.6mm FR-4	106x46 mm	1	10,85 €	10,85 €
Q1	12MHz	HC49/S	1	0,09 €	0,09 €
Q2	16MHz	HC49/S	1	0,09 €	0,09 €
R1, R2	1k 1/4W	0207/7	2	0,04 €	0,08 €
Horas de montaje	Hora de montador	Horas / hombre	5	20€	100€
Horas de ajuste	Hora de montador	Horas / hombre	5	20€	100€

## 2.9. Botones y USB

Parte	Valor	Encapsulado/ medidas	Cantidad	Precio ud	Precio
C1, C2, C3, C4	1uF 25V	E2-5	4	0,08 €	0,32 €
C5, C6, C7, C8	100nF 25V	C5B3	4	0,06 €	0,24 €
IC1, IC3	4013D	SO14	2	0,25 €	0,50 €
IC2, IC4	7414D	SO14	2	0,26 €	0,52 €
Tira de pines JP1	Pin Header Hembra 2.54 mm 3A	1X08	1	0,06 €	0,06 €
Tira de pines JP2	Pin Header Hembra 2.54 mm 3A	1X07	1	0,06 €	0,06 €
Tira de pines JP3	Pin Header Hembra 2.54 mm 3A	1X04	1	0,05 €	0,05 €
PCB	Dos capas - 35u Cu, 1.6mm FR-4	148x47 mm	1	12 €	11,93 €
R1, R2, R3, R4, R5, R6, R7, R8, R9, R10	10k 1/4W	0207/10	10	0,04 €	0,40 €
R11, R12	1K 1/4W	0207/10	2	0,04 €	0,08 €
R15	1k 1/4W	0207/10	1	0,04 €	0,04 €
S1, S2	Encoder	B3F-10XX	2	0,52 €	1,04 €
SG2	F/CM12P	F/CM12P	1	0,25 €	0,25 €
Encoder SW1, SW4	EC12E_SW	ALPS_EC12E_SW	2	0,07 €	0,14 €
T1	BC547	TO92-EBC	1	0,15 €	0,15 €
X1	KUSBVX-BS1N	KUSBVX-BS1N	1	0,44 €	0,44 €
Horas de montaje	Hora de montador	Horas / hombre	5	20€	100€
Horas de ajuste	Hora de montador	Horas / hombre	5	20€	100€

## 2.10. Control de temperatura

Parte	Valor	Encapsulado/ medidas	Cantidad	Precio ud	Precio
C1, C2	100nF 25V	C5B5	2	0,10 €	0,20 €
D1, D2	1N4004	DO41-10	2	0,07 €	0,14 €
Ventilador FAN1, FAN2, FAN3, FAN4, NTCPUENTE, NTCRECT, OVERTEMP, VCC	JST Header Hembra 2.54 mm 2A	1x02	8	0,12 €	0,96 €
IC1	TINY84-PU	DIL14	1	1,15 €	1,15 €
IC2	78L05Z	TO92	1	0,17 €	0,17 €
KK1, KK2	SK95-2M3	SK95-2M3	2	0,45 €	0,90 €
PCB	Dos capas - 35u Cu, 1.6mm FR-4	57x60 mm	1	7,54 €	7,54 €
Q1, Q2	IRFZ44	TO220BV	2	0,24 €	0,48 €
R1, R3	1k 1/4W	0207/10	2	0,03 €	0,06 €
R2, R4	10k 1/4W	0207/10	2	0,03 €	0,06 €
R5	10k 1% 1/4W	0207/10	1	0,05 €	0,05 €
R6	10k !% 1/4W	0207/10	1	0,05 €	0,05 €
SV1, SV2	Pin Header Macho 2.54 mm 3A	2x05	2	0,11 €	0,22 €
Horas de montaje	Hora de montador	Horas / hombre	5	20€	100€
Horas de ajuste	Hora de montador	Horas / hombre	5	20€	100€

## 2.11. Etapa de Salida

Parte	Valor	Encapsulado/ medidas	Cantidad	Precio ud	Precio
Terminal Faston GND1, GND2, GND3, VOUT1, VOUT2, VOUT3	Faston Terminal macho 6.3mm 16A	W237-102	6	0,11 €	0,66 €
CTRLRELE, VOUT	JST Header Hembra 2.54 mm 2A	1x02	2	0,11 €	0,22 €
K1	SLA-12VDC	SLA-12VDC	1	0,95 €	0,95 €
PCB	Dos capas - 35u Cu, 1.6mm FR-4	31x122 mm	1	10,49 €	10,49 €
Horas de montaje		Horas / hombre	5	20€	100€
Horas de ajuste		Horas / hombre	5	20€	100€

## 2.12. Piezas Auxiliares y pequeño material

Parte	Valor/medidas	Cantidad	Precio ud	Precio
Panel Frontal PLA	Kg PLA	0.116 Kg	0,03 €	3,48 €
Panel Trasero PLA	Kg PLA	0.109 Kg	0,03 €	3,27 €
Fuente de alimentación 12V1A	12V 1A	1	2,76 €	2,76 €
Fuente de alimentación 12V2A	12V 2A	1	4,95 €	4,95 €
Conector AC 3 terminales	C14	1	0,62 €	0,62 €
Terminal 4 mm	4 mm 30A	2	2,35 €	4,70 €
Interruptor AC 10A		1	0,67 €	0,67 €
Cubrebotón Encoders		2	0,57 €	1,14 €
Pantalla Nextion 3.2"		1	16,34 €	16,34 €
Ventiladores	Delta FFB0412SVHN	4	2,90 €	11,60 €
Terminales Faston Hembra	Faston Hembra 6.3 mm	25	0,14 €	3,50 €
Cable 2.5 mm (m)	30 mm	2	4 €	8,00 €
Cable 1.5 mm (m)	30 mm	3	2 €	6,00 €
PicKit 3		1	12 €	12,00 €
Tapa de acero inferior	295 x 280 mm	1	7 €	7,00 €
Tapa de acero superior	446 x 280 mm	1	9 €	9,00 €
Tornillos M3 5 mm	M3x5	11	0,25 €	2,75 €
Tornillos M3 10 mm	M3x10	11	0,21 €	2,31 €
Tuercas M3	M3	11	0,11 €	1,21 €
Separadores M3 5 mm	M3x5	11	0,34 €	3,74 €
Tirafondos M3 8 mm	M3x8	8	0,26 €	2,08 €
NTC B3950	10k	2	1,03 €	2,06 €
Horas de montaje	Horas / hombre	10	20€	200€
Horas de ajuste	Horas / hombre	10	20€	200€

### 2.13. Subtotal Materiales y elementos para la construcción del prototipo montados y ajustados

Parte	Precio
PCB Controlador PFC	210,92 €
PCB Entrada AC y Filtro EMI	325,13 €
PCB Convertidor PFC	446,70 €
PCB Convertidor DC-DC	466,58 €
PCB Control del convertidor	416,02 €
Driver FB x2	426,62 €
Driver SR	209,34 €
PCB Control de temperatura	211,98 €
PCB Control Panel Frontal	214,15 €
PCB Botones y USB	216,22 €
Etapa de Salida	212,32 €
Piezas auxiliares y pequeño material	508,07 €

**TOTAL DEL CAPÍTULO.....3864,05 €**

**El total del capítulo asciende a TRES MIIl OCHOCIENTOS SESENTA Y CUATRO EUROS con CINCO CÉNTIMOS.**

### 2.14. Subtotal Mano de obra de desarrollo

Tipo de trabajo	Cantidad	Precio ud	Subtotal
Horas de investigación y diseño	300 h	30€	9000€
Horas de programación	230 h	30€	6900€
Horas de depuración	40 h	30€	1200€
Horas de documentación	110h	20€	2200€

**TOTAL DEL CAPÍTULO.....19300 €**

**El total del capítulo asciende a DIECINUEVE MIL TRESCIENTOS EUROS.**

### 3. Total costes directos

Categoría	Subtotal
Materiales y elementos para la construcción del prototipo	3864,05€
Mano de obra de desarrollo	19300€

**TOTAL PARCIAL.....23164,05 €**

**El total del capítulo asciende a VEINTITRÉS MIL CIENTO SESENTA Y CUATRO EUROS con CINCO CÉNTIMOS.**

### 4. Coste general

Categoría	Porcentaje	Subtotal
Subtotal CD		23164,05€
Gastos generales (Costes indirectos)	15% (CD)	3474,61€
Beneficio industrial	6% (CD + CI)	1598,32€

**COSTE GENERAL.....28236,98 €**

**El total del capítulo asciende a VEINTIOCHO MIL DOSCIENTOS TREINTA Y SEIS EUROS con NOVENTA Y OCHO CÉNTIMOS.**

### 5. Total libre de impuestos

Categoría	Subtotal
Coste general	28236,98€

**TOTAL LIBRE DE IMPUESTOS.....28236,98 €**

**El total del prototipo, libre de impuestos, asciende a VEINTIOCHO MIL DOSCIENTOS TREINTA Y SEIS EUROS con NOVENTA Y OCHO CÉNTIMOS.**

### 6. Total final

Categoría	Porcentaje	Subtotal
Total libre de impuestos		28236,98€
IVA	21%	5929,77€

**TOTAL FINAL.....34166,75 €**

**El total del prototipo asciende a TREINTA Y CUATRO MIL CIENTO SESENTA Y SEIS EUROS con SETENTA Y CINCO CÉNTIMOS.**

Firmado, a 31 de agosto de 2019, Logroño



Alberto Otero Olavarrieta



# Bibliografía



A continuación, se proporciona una relación de las páginas WEB donde se ha encontrado la información más relevante consultada para este TFG, y que ha servido de complemento a la proporcionada dentro de los procesos formativos asociados a las asignaturas de los estudios de grado.

- Coilcraft. Common Mode Filter Design Guide.

[https://www.coilcraft.com/pdfs/doc191\\_CMfiltDesign.pdf](https://www.coilcraft.com/pdfs/doc191_CMfiltDesign.pdf)

- J. Jiraprasertwong, C. Jettanasen. Practical Design of a Passive EMI Filter for Reduction of EMI Generation.

[http://www.iaeng.org/publication/IMECS2015/IMECS2015\\_pp639-642.pdf](http://www.iaeng.org/publication/IMECS2015/IMECS2015_pp639-642.pdf)

- Nidhal Y. Nasser. Practical Approach in Designing Conducted EMI Filter to Mitigate Common Mode and Differential Mode Noises in SMPS.

<https://pdfs.semanticscholar.org/e458/b08b9bb991dd85110a14b243de267050eb4c.pdf>

- ON Semiconductor. Power Factor Correction (PFC) Handbook.

<https://www.onsemi.com/pub/Collateral/HBD853-D.PDF>

- Infineon Technologies. PFC boost converter design guide.

[https://www.infineon.com/dgdl/Infineon-ApplicationNote\\_PFCCMBoostConverterDesignGuide-AN-v02\\_00-EN.pdf?fileId=5546d4624a56eed8014a62c75a923b05](https://www.infineon.com/dgdl/Infineon-ApplicationNote_PFCCMBoostConverterDesignGuide-AN-v02_00-EN.pdf?fileId=5546d4624a56eed8014a62c75a923b05)

- Infineon Technologies. EVALPFC-300W-ICE3PCS01G 300 W PFC Evaluation Board with CCM PFC controller ICE3PCS01G.

[https://www.infineon.com/dgdl/Infineon-EVALPFC-300W-ICE3PCS01G-RD-v02\\_00-EN.pdf?fileId=db3a30433580b37101358a4e00996b86](https://www.infineon.com/dgdl/Infineon-EVALPFC-300W-ICE3PCS01G-RD-v02_00-EN.pdf?fileId=db3a30433580b37101358a4e00996b86)

- Elliott Sound Products. [www.sound-au.com](http://www.sound-au.com). Passive Power Factor Correction.

<https://sound-au.com/lamps/pfc-passive.html>

- Texas Instruments. MOSFET power losses and how they affect power-supply efficiency.

<http://www.ti.com/lit/an/slyt664/slyt664.pdf>

- Texas Instruments. Phase-Shifted Full Bridge DC/DC Power Converter Design Guide.

<http://www.ti.com/lit/ug/tidu248/tidu248.pdf>

- Texas Instruments. A Design Review of a Full-Featured 350-W Offline Power Converter.

<http://www.tij.co.jp/jp/lit/ug/tidu186/tidu186.pdf>

- STMicroelectronics. STEVAL-ISA172V2: 2 kW fully digital AC - DC power supply (D-SMPS) evaluation board. AN4856 Application note.

[https://www.st.com/content/ccc/resource/technical/document/application\\_note/group0/36/57/65/f9/b5/04/41/37/DM00284856/files/DM00284856.pdf/jcr:content/translations/en.DM00284856.pdf](https://www.st.com/content/ccc/resource/technical/document/application_note/group0/36/57/65/f9/b5/04/41/37/DM00284856/files/DM00284856.pdf/jcr:content/translations/en.DM00284856.pdf)

- Texas Instruments. UCC28950 600-W, Phase-Shifted, Full-Bridge Application Report.

<http://www.ti.com/lit/an/slva560c/slva560c.pdf>

- Infineon Technologies. ZVS Phase Shift Full Bridge CFD2 Optimized Design.

[https://www.infineon.com/dgdl/Infineon-ApplicationNote\\_EvaluationBoard\\_EVAL\\_2KW\\_ZVS\\_FB\\_CFD2+ZVS-AN-v01\\_00-EN.pdf?fileId=db3a30433f9a93b7013f9f582ddb1fd9](https://www.infineon.com/dgdl/Infineon-ApplicationNote_EvaluationBoard_EVAL_2KW_ZVS_FB_CFD2+ZVS-AN-v01_00-EN.pdf?fileId=db3a30433f9a93b7013f9f582ddb1fd9)

- Microchip Technology. Phase-Shifted Full-Bridge (PSFB) Quarter Brick DC/DC Converter Reference Design Using a dsPIC® DSC.

[http://ww1.microchip.com/downloads/en/AppNotes/01335A\\_Appnote.pdf](http://ww1.microchip.com/downloads/en/AppNotes/01335A_Appnote.pdf)

- STMicroelectronics. Topologies for switched mode power supplies.

[https://www.st.com/content/ccc/resource/technical/document/application\\_note/2a/8c/7f/9e/70/6b/43/62/CD00003910.pdf/files/CD00003910.pdf/jcr:content/translations/en.CD00003910.pdf](https://www.st.com/content/ccc/resource/technical/document/application_note/2a/8c/7f/9e/70/6b/43/62/CD00003910.pdf/files/CD00003910.pdf/jcr:content/translations/en.CD00003910.pdf)

- Biricha Digital Power. Step-by-Step Design Guide for Digital Peak Current Mode Control: A SingleChip Solution.

<https://www.biricha.com/uploads/8/9/8/0/89803127/step-by-step-design-guide-for-digital-peak-current-mode-control-a-single-chip-solution.pdf>