

UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN



PROYECTO FIN DE CARRERA

TITULO: DISEÑO Y FABRICACIÓN DE LOS SISTEMAS
DE CONTROL Y SINCRONISMO DIGITALES EN
RADARES DE IMAGEN EN BANDAS MILIMÉTRICAS

AUTOR: Alejandro Badolato Martín

AÑO: 2013

DEPARTAMENTO DE SISTEMAS, SEÑALES Y RADIOPROGRAMACIONES

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE TELECOMUNICACIÓN
UNIVERSIDAD POLITÉCNICA DE MADRID

DISEÑO Y FABRICACIÓN DE LOS SISTEMAS DE
CONTROL Y SINCRONISMO DIGITALES EN RADARES
DE IMAGEN EN BANDAS MILIMÉTRICAS

PROYECTO FIN DE CARRERA

Autor:
Alejandro Badolato Martín

TÍTULO: DISEÑO Y FABRICACIÓN DE LOS SISTEMAS DE CONTROL Y SINCRONISMO DIGITALES EN RADARES DE IMAGEN EN BANDAS MILIMÉTRICAS

AUTOR: Alejandro Badolato Martín

DEPARTAMENTO: Señales, Sistemas y Radiocomunicaciones

MIEMBROS DEL TRIBUNAL CALIFICADOR

PRESIDENTE: D. ALBERTO ASENSIO LÓPEZ

VOCAL: D. JESÚS GRAJAL DE LA FUENTE

SECRETARIO: D. OMAR ARTEMI YESTE OJEDA

SUPLENTE: D. JAVIER GISMERO MENOYO

FECHA DE LECTURA:

CALIFICACIÓN:

RESUMEN DEL PROYECTO:

Gracias a los avances tecnológicos logrados en los últimos años en el campo de la radiofrecuencia, se pueden encontrar en el mercado componentes que trabajan en las bandas milimétrica (30-300 GHz) y submilimétrica (300 GHz - 3 THz). A dichas frecuencias se pueden desarrollar nuevas aplicaciones radar de muy alta resolución. Entre otras figuran la ayuda a la conducción o ACC (Adaptive Cruise Control), la asistencia al aterrizaje de aeronaves o la obtención de imágenes SAR/ISAR (Synthetic Aperture Radar / Inverse Synthetic Aperture Radar). En esta banda de frecuencias también es posible desarrollar sistemas capaces de detectar objetos ocultos en el interior de paquetes o maletas o incluso obtener imágenes de alta resolución.

El Grupo de Microondas y Radar de la ETSIT, en colaboración con la EET (Escola de Enxeñaría de Telecomunicación) de la Universidad de Vigo, está desarrollando un radar cuyo objetivo principal es tratar de que las personas que van a acceder a un recinto de seguridad lo hagan sin portar armas o explosivos ocultos bajo la ropa.

Dicho radar trabaja en la banda de los 300 GHz y es capaz de obtener una imagen radar tridimensional del objetivo en un tiempo inferior a 1 segundo con resolución inferior a 1.5 cm en cada uno de sus tres ejes.

Este proyecto fin de carrera tiene por cometido desarrollar algunos de los sistemas necesarios para que el citado radar sea operativo. Tales sistemas son el sistema de alimentación (véase el capítulo 2) y el sistema de control y sincronismo (véase el capítulo 5). Para un mayor detalle del alcance del proyecto véase 1.5.

PALABRAS CLAVE:

Radar Alta Resolución, radar de imagen, detección armas, detección explosivos, control de accesos, terahercio, radar 3D, radar tridimensional.

Índice general

1 Introducción.	1
1.1 Sistemas radar de ondas submilimétricas.	1
1.2 Escáner corporal <i>Provision2</i>	3
1.3 Escáner corporal de retrodispersión de rayos X.	4
1.4 Escáner corporal de ondas milimétricas en desarrollo por la UPM.	6
1.4.1 Introducción.	6
1.4.2 Sistemas que componen el escáner.	8
1.5 Alcance y descripción del proyecto.	9
2 Sistema de alimentación.	11
2.1 Introducción.	11
2.2 Diseño de fuente inicial.	12
2.2.1 Tarjeta de rectificación.	15
2.2.2 Tarjeta de regulación.	17
2.2.3 Caracterización de la fuente.	17
2.2.4 Lecciones aprendidas.	19
2.3 Diseño de fuente modular.	20
2.3.1 Rack.	21
2.3.2 Backplane.	22
2.3.3 Tarjetas de rectificación.	23
2.3.4 Tarjeta de regulación.	26
2.3.4.1 Análisis térmico.	35
2.3.4.2 Caracterización de la tarjeta.	38
2.3.5 Tarjeta controladora.	39
2.3.6 Software de monitorización y control.	40
3 Sistema de antenas.	43
3.1 Descripción del sistema de antenas.	43
3.1.1 Reflector principal.	52
3.1.2 Subreflector.	53
3.1.3 Configuración bifocal.	54
3.1.4 Espejo de exploración.	59
3.1.5 Reflector de alimentación.	60
3.1.6 Beam splitter.	61

ÍNDICE GENERAL

3.1.7	Bocinas.	62
3.1.8	Plano del objetivo.	63
3.1.9	Movimiento descrito por el sistema de antenas.	64
3.2	Subsistema de apuntamiento.	66
3.2.1	Simulación del subsistema de apuntamiento.	67
3.2.2	Motor de rotación.	69
3.2.2.1	Reductora.	70
3.2.2.2	Encoder.	71
3.2.2.3	Controlador.	72
3.2.2.4	Caracterización del motor.	74
3.2.3	Motor de elevación.	77
3.2.3.1	Encoder.	81
3.2.3.2	Controlador.	81
3.2.4	Referencias del subsistema de apuntamiento.	83
3.2.4.1	Referencia del motor de rotación.	84
3.2.4.2	Referencias del motor de elevación.	85
3.2.4.3	Referencia del plano del objetivo.	86
3.2.4.4	Apuntamiento del sistema en el plano del objetivo. . .	86
3.2.4.5	Función coordenadas del blanco.	87
3.2.4.6	Representación gráfica de blancos.	87
3.2.4.7	Sistema de coordenadas absolutas.	88
3.3	Caracterización del sistema de antenas.	89
4	Sistema de generación de señal de radiofrecuencia.	101
4.1	Introducción.	101
4.2	DDS Euvis.	101
5	Sistema de control y sincronismo.	105
5.1	Subsistema de sincronismo.	106
5.1.1	Generador de señales de reloj.	109
5.1.1.1	Señal de reloj de 10 MHz.	110
5.1.1.2	Señal de reloj de 2.4 GHz.	112
5.1.1.3	Señal de reloj de 500 MHz.	113
5.1.1.4	Señal de reloj de 40 MHz.	114
5.1.1.5	Tarjeta de generación de señales de reloj.	116
5.1.1.6	Caja de generación de señales.	120
5.1.2	Tarjeta de digitalización.	120
5.1.2.1	Capacidad de transferencia.	123
5.1.3	Tarjeta de inicio de captura.	123
5.2	Subsistema de control.	126
5.2.1	Configuración del ADC.	127
5.2.2	Configuración del motor de rotación.	128
5.2.3	Configuración del motor de elevación.	130
5.2.4	Configuración del generador de señal <i>trigger</i>	131

6 Mejoras propuestas.	133
7 Conclusiones.	137
Bibliografía	139
Apéndices	143
A Documentación del controlador del motor de rotación.	145
A.1 Modos de regulación	145
A.1.1 Control de corriente.	145
A.1.2 Control de velocidad.	145
A.1.3 Control de posición.	147
A.2 Control del motor de rotación mediante Epos Studio.	148
A.2.1 Introducción.	148
A.2.2 <i>Profile Velocity Mode</i>	149
A.2.3 <i>Profile Position Mode</i>	150
A.2.4 <i>Homing Mode</i>	151
A.2.5 <i>Device Control</i>	152
A.3 Calibración del motor.	152
A.4 Control del motor de rotación mediante API.	155
A.4.1 Introducción.	155
A.4.2 Instalación de la librería.	155
A.4.3 Comunicación con el controlador EPOS2.	156
A.4.4 Selección del modo de operación.	156
A.4.5 Excitación de los bobinados.	156
A.4.6 Inicio de la rotación.	157
B Documentación tarjeta de síntesis de señales de reloj.	159
B.1 Módulo MOD5270.	159
B.2 Programación del sintetizador mediante PIC.	162
B.3 Esquemas tarjeta de síntesis de señales de reloj.	182
B.4 Esquemas tarjeta acoplamiento MOD5270.	191
B.5 Código tarjeta de síntesis de señales de reloj.	196
C DDS Analog Devices.	205
C.1 Generador de señal de disparo.	210
D Documentación tarjeta digitalizadora.	215
D.1 Modos de disparo soportados.	215
D.1.1 Post-Trigger.	215
D.1.2 Pre-Trigger.	215
D.1.3 Middle-Trigger.	216
D.1.4 Delay-Trigger.	216
D.2 Submuestreo.	216

D.3	Modo de captura simple.	217
D.4	Modo de captura continuo.	217
D.5	Driver de la tarjeta.	219
D.5.1	<i>WD_Register_Card.</i>	220
D.5.2	<i>WD_AI_ContBufferSetup.</i>	220
D.5.3	<i>WD_AI_CH_ChangeParam.</i>	220
D.5.4	<i>WD_AI_Config.</i>	220
D.5.5	<i>WD_AI_Trig_Config.</i>	221
D.5.6	<i>WD_AI_AsyncDblBufferMode.</i>	221
D.5.7	<i>WD_AI_ContScanChannels.</i>	221
D.5.8	<i>WD_AI_AsyncDblBufferHalfReady.</i>	221
D.5.9	<i>WD_AI_AsyncClear.</i>	221
E	Documentación tarjeta de inicio de captura.	223
E.1	Esquemas tarjeta de inicio de captura.	223
E.2	Código tarjeta de inicio de captura.	234
F	Código Matlab.	243
F.1	Código para la determinación de las coordenadas del blanco.	243
F.2	Código para representar gráficamente los blancos.	246

Índice de figuras

1.1	Scanner corporal basado en ondas milimétricas.	3
1.2	Imágenes obtenidas mediante scanner de ondas milimétricas.	3
1.3	Movimiento del sistema de antenas del scanner de ondas milimétricas. . .	4
1.4	Escáner corporal basado en retrodispersión de rayos X.	5
1.5	Imágenes obtenidas mediante scanner de rayos X.	6
1.6	Escáner corporal diseñado.	7
1.7	Comparativa entre sistemas.	8
1.8	Sistemas del escáner corporal.	9
2.1	Rizado de la fuente de alimentación Agilent E3631A a 5V sin carga. . . .	12
2.2	Fuente de alimentación inicial.	13
2.3	Esquema de la fuente.	14
2.4	Esquema de rectificado del transformador B.	16
2.5	Simulación del circuito de rectificación.	16
2.6	Fuente lineal.	17
2.7	Tensión de salida del circuito rectificador del devanado A1.	18
2.8	Medida tensión de rizado 7 V	18
2.9	Medida tensión de rizado 5 V	18
2.10	Fuente modular.	20
2.11	Fuente modular.	21
2.12	Rack de la fuente modular.	21
2.13	Rack de la fuente modular.	22
2.14	Estructura del <i>Backplane</i>	22
2.15	<i>Backplane</i> de rectificación.	23
2.16	<i>Backplane</i> de regulación.	23
2.17	Tarjeta de rectificación #1.	24
2.18	Conexionado en serie de los devanados secundarios.	25
2.19	Tarjeta de rectificación #3.	25
2.20	Tarjeta de regulación.	26
2.21	Circuito de conmutación de la tarjeta de regulación.	27
2.22	Rechazo a la tensión de rizado del regulador LT1083.	28
2.23	Circuito de regulación simplificado I.	28
2.24	Circuito de regulación simplificado II.	29
2.25	Señales de entrada y salida del comparador U204.	30
2.26	Circuito de regulación simplificado IV.	30

ÍNDICE DE FIGURAS

2.27	Círculo de filtrado de la tarjeta de regulación	31
2.28	Esquema de simulación del circuito de regulación	32
2.29	Resultado de la simulación del circuito de regulación I	33
2.30	Resultado de la simulación del circuito de regulación II	34
2.31	Potencia disipada por los reguladores	37
2.32	Tensión de rizado medida de la fuente de alimentación	38
2.33	Tarjeta controladora	39
3.1	Sistema de antenas	43
3.2	Sistema de antenas	44
3.3	Volumen explorado por el sistema de antenas	45
3.4	Espejo de exploración	46
3.5	Eje de rotación con $\gamma = 0^\circ$	46
3.6	Eje de rotación con $\gamma = -6.8^\circ$	46
3.7	Detalle del espejo exploración	47
3.8	Exploración del plano del objetivo	47
3.9	Sistema de coordenadas de la antena	47
3.10	Elipsoide que modela el reflector principal	49
3.11	Vista de planta del sistema de antenas	49
3.12	Trayectoria descrita por el <i>spot</i> en el plano del objetivo	51
3.13	Reflector principal	52
3.14	Subreflector	53
3.15	Orientación del plano focal	55
3.16	Distancia de los <i>spots</i> al anillo focal	56
3.17	Desfase medio RMS según distancia al foco secundario	57
3.18	Valor del campo relativo en 8 puntos de la elipse central	57
3.19	Tamaño del <i>spot</i> en 8 puntos de la elipse central	58
3.20	Valor del campo relativo en 8 puntos de la elipse superior	58
3.21	Tamaño del <i>spot</i> en 8 puntos de la elipse superior	59
3.22	Espejo de exploración	59
3.23	Estructura soporte del espejo de exploración	60
3.24	Reflector de alimentación	61
3.25	<i>Beamsplitter</i>	61
3.26	Descripción del <i>Beamsplitter</i>	62
3.27	Bocinas	62
3.28	Excitación de modos de la bocina tipo <i>Potter</i>	63
3.29	Orientación del eje X_T del plano del objetivo	63
3.30	Movimiento del espejo de exploración	64
3.31	Diagrama de flujo del sistema de sincronismo	65
3.32	Trayectoria descrita por el <i>spot</i> en el plano del objetivo	66
3.33	Estructura soporte del espejo de exploración	67
3.34	Apuntamiento de rayos en el plano focal	68
3.35	Conexiones entre los elementos del sistema de rotación	69
3.36	Reductora PLE 40-4	70

3.37	Vida útil del lubricante de la reductora.	71
3.38	Encoder del motor de rotación.	72
3.39	Controlador del motor de rotación.	73
3.40	Diccionario de objetos del controlador EPOS2 70/10.	73
3.41	Filtrado de los pulsos del encoder.	75
3.42	Función densidad de probabilidad de la velocidad instantánea.	75
3.43	Velocidad media de rotación.	76
3.44	Movimiento angular de elevación.	77
3.45	Movimiento angular de elevación.	78
3.46	Subsistema de elevación.	78
3.47	Motor de elevación.	79
3.48	Motor por pasos.	79
3.49	Motor por pasos con polarización alternada.	80
3.50	Husillo del motor de elevación.	80
3.51	Curvas de carga de los motores de elevación.	81
3.52	Controlador del motor de elevación.	81
3.53	GUI del controlador Haydon.	82
3.54	Maniobra de extensión.	83
3.55	Sensor inductivo.	84
3.56	Referencia externa para la determinación de ϕ	84
3.57	Chapas de los fines de carrera.	85
3.58	Sensores inductivos.	85
3.59	Puntero láser para referenciar el plano del objetivo.	86
3.60	Sistema de coordenadas absolutas.	88
3.61	Sistema de coordenadas absolutas.	90
3.62	Matriz de bolas capturada.	91
3.63	Detalle de la bola central de la matriz.	92
3.64	Detalle de la bola central de la matriz.	92
3.65	Perfiles de potencia vertical y horizontal del spot.	93
3.66	Medida de la bola de 5mm de la fila 1, columna 6.	93
3.67	Medida de la bola de 5mm de la fila 2, columna 5.	94
3.68	Medida de la bola de 5mm de la fila 3, columna 6.	94
3.69	Medida de la bola de 5mm de la fila 5, columna 2.	95
3.70	Medida de la bola de 5mm de la fila 5, columna 6.	95
3.71	Medida de la bola de 5mm de la fila 7, columna 4.	96
3.72	Medida de la bola de 5mm de la fila 9, columna 2.	96
3.73	Medida de la bola de 5mm de la fila 9, columna 6.	97
3.74	Medida de la bola de 5mm de la fila 10, columna 1.	97
3.75	Medida de la bola de 5mm de la fila 12, columna 3.	98
3.76	Anchura del spot.	98
3.77	Altura del spot.	98
3.78	Matriz a 8 m del foco primario de la antena.	99
3.79	Matriz a 8.22 m del foco primario de la antena.	99

3.80	Matriz a 7.85 m del foco primario de la antena.	99
4.1	<i>Direct Digital Synthesizer DSM303B.</i>	102
4.2	<i>Direct Digital Synthesizer DSM303B.</i>	103
4.3	<i>Fuente conmutada PTH12050W.</i>	104
4.4	Fuentes retiradas de la tarjeta.	104
5.1	Diagrama del sistema de control y sincronismo.	105
5.2	Señal de batido capturada.	106
5.3	FFT de la señal de batido.	106
5.4	Atenuador de ajuste manual a la entrada del triplicador.	107
5.5	Movimiento del espejo de exploración.	108
5.6	Diagrama de flujo del sistema de sincronismo.	109
5.7	Relojes del sistema.	110
5.8	Cristal de cuarzo de 10 MHz.	110
5.9	Ruido de fase.	111
5.10	Generación de señales de referencia.	112
5.11	Ruido de fase.	113
5.12	Placa de evaluación para sintetizadores Synergy.	113
5.13	Software para la programación de sintetizadores Synergy.	114
5.14	Programador de la tarjeta de evaluación.	114
5.15	Ruido de fase.	115
5.16	Respuesta en frecuencia del filtro SAW.	116
5.17	Tarjeta de síntesis generación de señales de reloj.	118
5.18	Mecanización de la lámina de latón.	118
5.19	PCB para la inserción del módulo MOD5270.	119
5.20	Caja de generación de señales I.	120
5.21	Caja de generación de señales II.	121
5.22	Tarjeta ADC.	121
5.23	Diagrama funcional del ADC.	122
5.24	Diagrama de bloques del circuito de entrada de un canal.	123
5.25	Tarjeta de inicio de captura.	124
5.26	Captura de osciloscopio de pulsos de contadores y señal de disparo.	126
5.27	Formulario principal.	127
5.28	Formulario de configuración del ADC.	128
5.29	Formulario de configuración del ADC.	128
5.30	Formulario de control y configuración del motor de rotación.	129
5.31	Formulario de control y configuración del motor de elevación.	130
5.32	Formulario de configuración de la señal <i>trigger</i>	131
A.1	Controlador PI de corriente.	145
A.2	Controlador PI de velocidad.	146
A.3	Respuesta del filtro de promediado de velocidad.	146
A.4	Controlador PID de posición.	147

A.5	<i>EPOS2 Studio 2.00.</i>	148
A.6	Profile Velocity Mode.	149
A.7	Perfil de aceleración en \sin^2 .	149
A.8	Profile Position Mode.	150
A.9	Homing Mode.	151
A.10	Device Control.	152
A.11	<i>Expert Tuning:</i> Etapa de cascada.	153
A.12	<i>Expert Tuning:</i> Etapa de identificación.	153
A.13	<i>Expert Tuning:</i> Etapa de parametrización.	153
A.14	<i>Expert Tuning:</i> Etapa de verificación.	154
B.1	Arquitectura del microcontrolador MC5270.	160
B.2	Módulo 5270.	161
C.1	Placa de evaluación AD9958/PCBZ.	205
C.2	Forma de onda del barrido lineal en AD9885.	206
C.3	GUI de la placa de evaluación AD9958/PCBZ.	208
C.4	DDS AD9958.	208
C.5	Esquema del conector U2 de la placa de evaluación.	209
C.6	Buffer de salida de la tarjeta de evaluación.	210
C.7	Diagrama de bloques de los temporizadores programables.	211
C.8	Conector de salida de la señal de <i>trigger</i> .	211
C.9	Aplicación para configurar el <i>trigger</i> digital.	213
D.1	Modo de captura <i>post-trigger</i> .	215
D.2	Modo de captura <i>pre-trigger</i> .	216
D.3	Modo de captura <i>middle-trigger</i> .	216
D.4	Modo de captura <i>delay-trigger</i> .	216
D.5	Submuestreo aplicado según el valor de scan interval.	217
D.6	Software <i>DAQ-PILOT</i> .	219
D.7	Tarjeta ADC.	220

Índice de Tablas

1.1	Sistemas radar en banda W.	2
1.2	Sistemas para la obtención de imágenes en ondas milimétricas y submilimétricas.	2
2.1	Tensiones de la fuente de alimentación.	11
2.2	Transformadores toroidales.	13
2.3	Transformador A.	15
2.4	Transformador B.	15
2.5	Transformador C.	15
2.6	Tensiones de la fuente modular.	20
2.7	Descripción de las medidas.	33
2.8	Eficiencia de las fuentes.	35
2.9	Modelo térmico del regulador.	35
2.10	Temperaturas del regulador.	36
2.11	Funciones de la tarjeta controladora.	40
3.1	Elementos del sistema de antenas.	44
3.2	Descripción de la superficie del reflector principal.	53
3.3	Descripción de la superficie del subreflector.	54
3.4	Descripción de la superficie del reflector de alimentación.	61
3.5	Valor de las constantes de las ecuaciones 3.12 y 3.13.	69
3.6	Características del motor de rotación.	70
3.7	Parámetros de entrada de la función de posicionamiento de blancos.	87
3.8	Medida de matriz de bolas de 5mm.	89
5.1	Especificaciones técnicas del ADC.	122
5.2	Sistemas configurables desde la aplicación del sistema.	126
A.1	Parámetros de calibración del controlador.	154
A.2	Modos de operación del controlador EPOS2 70/10.	157
C.1	Parámetros del barrido lineal en frecuencia.	206

Capítulo 1

Introducción.

1.1. Sistemas radar de ondas milimétricas.

En la banda de frecuencias situada entre 30 y 300 GHz la longitud de onda en el vacío es $1mm \leq \lambda \leq 1cm$. Hasta hace poco tiempo, en esta zona del espectro no se habían desarrollado aplicaciones debido a la escasa variedad de circuitos capaces de trabajar a tales frecuencias. En la actualidad comienza a existir tecnología comercial que permite diseñar nuevas aplicaciones en dicho rango de frecuencias según se muestra en la tabla 1.1 [1]. En estas frecuencias, los sistemas radiantes son de reducidas dimensiones y se puede transmitir un elevado ancho de banda. Aprovechando estas propiedades se están desarrollando nuevas aplicaciones radar tales como la ayuda a la conducción o ACC (*Adaptive Cruise Control*) [2] [3], la asistencia al aterrizaje de aeronaves [4] o la obtención de imágenes SAR/ISAR (*Inverse Synthetic Aperture Radar*) de alta resolución [5] [6] [7].

Además, en este margen de frecuencias las ondas tiene una gran capacidad para penetrar tejidos y una gran variedad de materiales. Por ello, otras aplicaciones a estas frecuencias son los sistemas radar de imagen tales como los mostrados en la tabla 1.2 que permiten realizar inspecciones de objetos de forma no intrusiva. Actualmente existen en el mercado sistemas capaces de explorar el torso de una persona con el fin de detectar si porta armas o explosivos ocultos bajo la ropa [8] [9] [10]. Como alternativa a estos sistemas, también existen dispositivos basados en la retrodispersión de rayos X [11] [12]. A continuación se detallan las características de cada uno de ellos.

1. Introducción.

	f_c (GHz)	Forma de onda	B(MHz)	ΔR (m)	Aplicación
Huggard <i>et al</i> [23]	94.0275	Diente de sierra	-	16-30	Perfilado de nubes
Guldogan <i>et al</i> [24]	77	Diente de sierra	150	1	registros Micro-Doppler
Brooker <i>et al</i> [25]	94	Diente de sierra	250	1.2	Industria y minería
Rangwala <i>et al</i> [26]	94.75	Diente de sierra	500	0.3	Aterrizaje asistido para helicópteros
Johannes <i>et al</i> [27]	94	Diente de sierra	1000	0.15	Imágenes SAR
Van Caekenberghue <i>et al</i> [28]	77	OFDM	2000	0.075	Sistema automático de aterrizaje
Essen <i>et al</i> [29]	94	Diente de sierra/SF	4000/800	0.035/19	Imágenes ISAR
Hantscher <i>et al</i> [30]	97.5	-	3000	0.05	Controles de seguridad
Kapilevich <i>et al</i> [31]	94	Diente de sierra	6000	0.025	Detección Standoff

Tabla 1.1: Sistemas radar en banda W.

	f_c (GHz)	Tipo	B (GHz)	ΔR (mm)	Resolución transversal(mm)	Distancia (m)	t_{adq} (s)	Campo de visión (m^2)
Gu <i>et al</i> [32]	205	Transceptor basado en VNA	10	15	7(H), 4(V)	4.5	<3	2x0.6
Song <i>et al</i> [33]	200	Radar CW	-	15	0.85	0.135		0.4x0.2
Goyette <i>et al</i> [34]	1560	Sistema optoelectrónico	-	-	15	2.54	0.5	0.51x0.51
Cooper <i>et al</i> [35]	676.5	Radar CW-LFM	29	5.2	10	25	<1	0.4x0.4
Sheen <i>et al</i> [36]	350	Radar CW-LFM	9.6	15.6	10	5	10	1.5x2.5
Weg <i>et al</i> [37]	300	Radar CW-LFM	90	16.7	8	0.5-1.5	9	0.2x0.3
Weg <i>et al</i> [37]	645	Radar CW-LFM	10	15	4	0.75-1.5	9	0.2x0.3

Tabla 1.2: Sistemas de imagen en ondas milimétricas y submilimétricas.

1.2. Escáner corporal *ProVision2*.

En muchos aeropuertos internacionales existen dispositivos de ondas milimétricas similares al mostrado en la figura 1.1, que garantizan un acceso seguro a las zonas de embarque. El equipo mostrado es el modelo Provision2 de la marca L-3 SDS (*Security % Detection Systems*) [13].



Figura 1.1: Scanner corporal basado en ondas milimétricas.

En la figura 1.2 se muestra un ejemplo de las imágenes obtenidas con este tipo de sistemas.

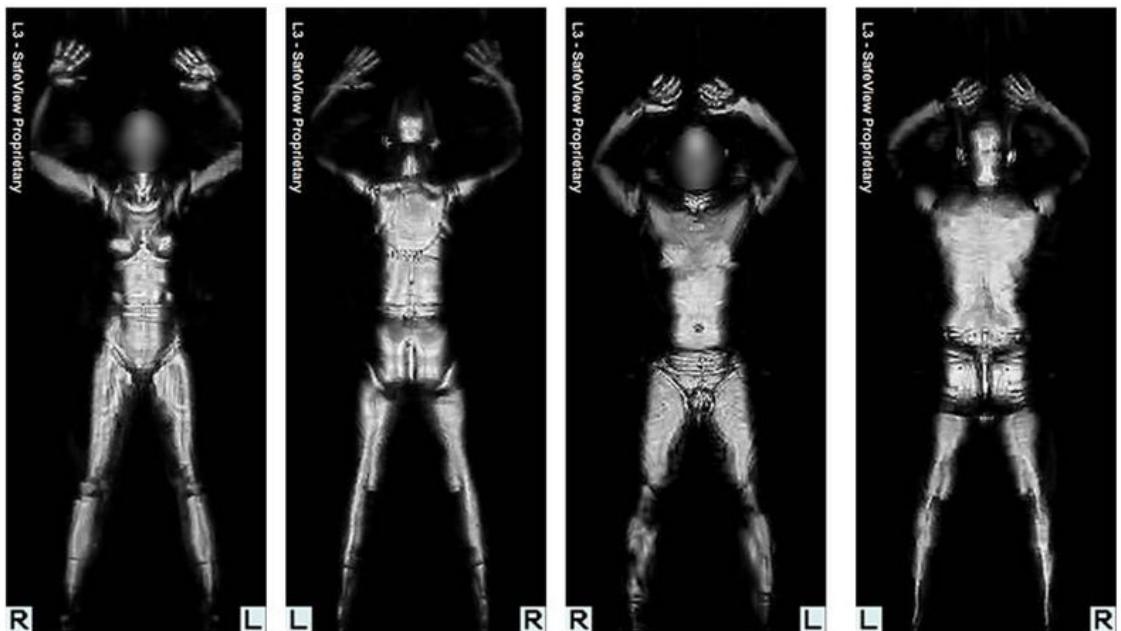


Figura 1.2: Imágenes obtenidas mediante scanner de ondas milimétricas.

En estos sistemas un array de antenas rota alrededor del cuerpo. Cada antena realiza funciones de transmisión y recepción que exploran el cuerpo radialmente a una altura determinada. El principio de funcionamiento puede apreciarse en la figura 1.3.

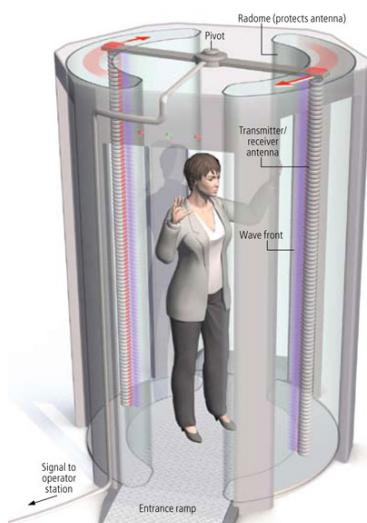


Figura 1.3: Movimiento del sistema de antenas del scanner de ondas milimétricas.

El tiempo mínimo requerido para obtener una imagen de 360 grados con estos dispositivos es de 8 segundos (2 segundos para escanear el cuerpo y 6 segundos para el posterior procesado) [13].

1.3. Escáner corporal de retrodispersión de rayos X.

Otro tipo de equipos que también están instalados en los principales aeropuertos internacionales para controlar un acceso seguro a las zonas de embarque, utiliza una técnica basada en la retrodispersión de rayos X [12] para detectar posibles objetos ocultos.

La longitud de onda de los rayos X es del orden de nanómetros y, por lo tanto, comparable con el tamaño de los átomos que componen la materia. De tal forma, los rayos X tienen la capacidad de interactuar con dichos átomos favoreciendo el intercambio de electrones de los mismos con otros átomos y formando de esta manera iones (átomos con carga). Estos iones pueden provocar cambios en las estructuras de las moléculas y generar lesiones celulares a nivel del ADN (Ácido Desoxirribonucleico) pudiendo llegar a provocar la aparición de cáncer.

No obstante, los sistemas de retrodispersión de rayos X emiten dosis extremadamente bajas comparadas con las pruebas médicas de radiodiagnóstico tales como radiografías o TACs (Tomografía Axial Computarizada). En estos sistemas lo que se analiza no es

la radiación que ha atravesado el cuerpo (como ocurre en un proceso radiográfico) sino la radiación reflejada principalmente en la superficie corporal.

A diferencia de los sistemas basados en ondas milimétricas, cuyas imágenes son tridimensionales, la imagen obtenida mediante esta técnica es bidimensional.

El escáner de rayos X obtiene una imagen anterior y posterior del cuerpo gracias a los detectores instalados a ambos lados de la persona a explorar según se muestra en la figura 1.4.

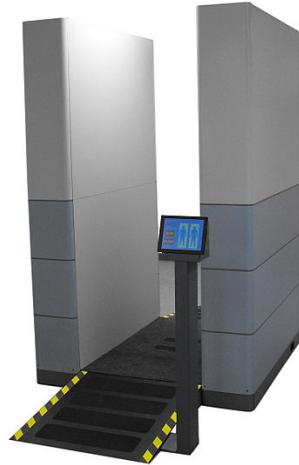


Figura 1.4: Escáner corporal basado en retrodispersión de rayos X.

Como se aprecia en la figura 1.5, las imágenes obtenidas mediante esta técnica muestran con mayor detalle la superficie corporal y disminuyen significativamente las falsas alarmas que se producen en los sistemas milimétricos por pliegues en la ropa, botones, e incluso las gotas de sudor.

La normativa ANSI/HPS N43.17 (American National Standard Institute y Health Physics Society respectivamente) limita a $0.25 \mu\text{Sieverts}$ la dosis máxima efectiva a la que puede ser sometida una persona durante un proceso de barrido con este tipo de sistemas [14]. Dicha dosis es similar a la dosis recibida por efecto de la radiación cósmica durante 2 minutos de vuelo [15] [16].

El tiempo de adquisición de una imagen mediante esta técnica es aproximadamente de 30 segundos. Durante este periodo se realizan 10 exploraciones para, posteriormente, realizar un proceso de integración. En cada una de las exploraciones la persona a escanear es sometida a una dosis de $0.02 \mu\text{Sieverts}$ [17].

A pesar de las bajas dosis de radiación emitidas, cuyos efectos no son dañinos para la salud según informes de la Unión Europea [18], la TSA (Transportation Security Administration de Estados Unidos) está trasladando dichos equipos a aeropuertos de menor tráfico y sustituyendo éstos por escáneres milimétricos [19].

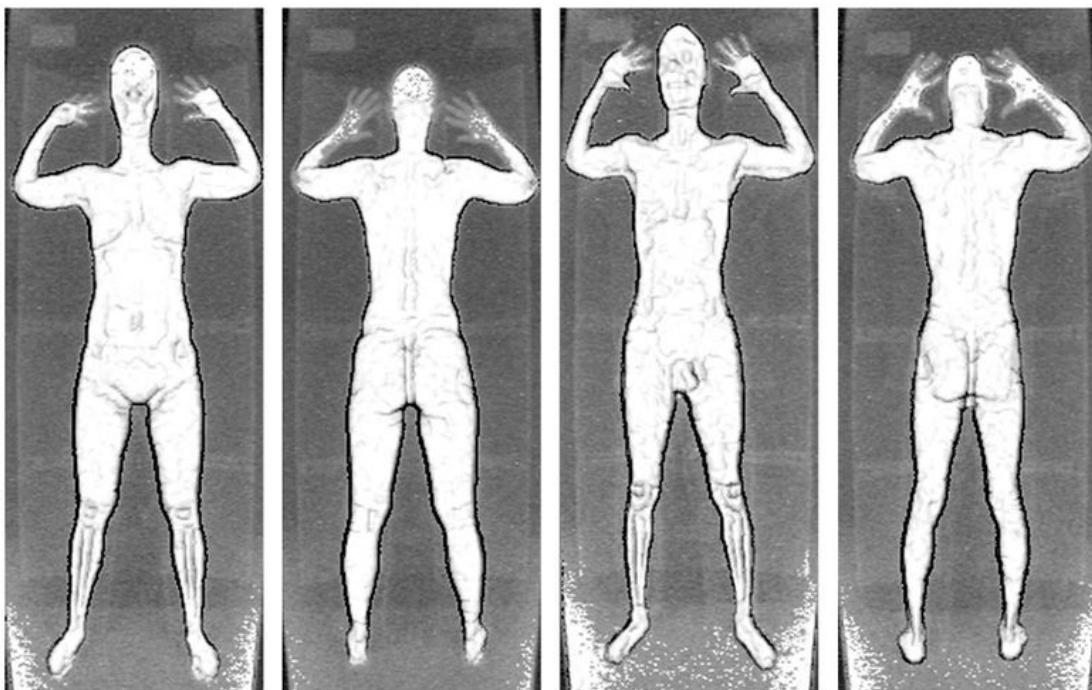


Figura 1.5: Imágenes obtenidas mediante scanner de rayos X.

1.4. Escáner corporal de ondas milimétricas en desarrollo por la UPM.

1.4.1. Introducción.

El Grupo de Microondas y Radar perteneciente al departamento de Señales Sistemas y Radiocomunicaciones está diseñando un escáner corporal de ondas milimétricas en la banda de los 300 GHz.

A diferencia de los equipos presentados en el apartado 1.2, el sistema propuesto pretende obtener una imagen corporal mediante un único sistema radiante ubicado a 8 metros del objeto que se va a explorar. El sistema pretende lograr imágenes corporales tridimensionales en tiempos cercanos al segundo.

El escáner tendrá capacidad para explorar un área de 50 cm de ancho por 1 metro de alto ubicada a 65 cm del suelo. Dicha área abarca aproximadamente la zona comprendida entre la rodilla y el cuello del torso de una persona adulta. Mostrará una imagen tridimensional al operador cuya resolución será inferior a 1.5 cm en cada uno de sus tres ejes ortogonales. En la figura 1.6 se muestra la parte superior de la estructura que incorpora todos los componentes del escáner a excepción de la interfaz HMI (Human Machine Interface) la cual se compone de ratón, teclado y pantalla.

En la parte inferior de la estructura quedan ocultos parte de los sistemas de radiofrecuencia, el sistema de alimentación (véase el capítulo 2), la caja de sincronismo (véase el apartado 5.1), la caja de generación de señales (véase el apartado 5.1.1.6), los controladores de los motores de rotación y elevación (véase los apartados 3.2.2.3 y 3.2.3.2 respectivamente), el ordenador que alberga el ADC (Analog to Digital Converter) (véase la sección 5.1.2) y la aplicación del subsistema de control (véase el capítulo 5).

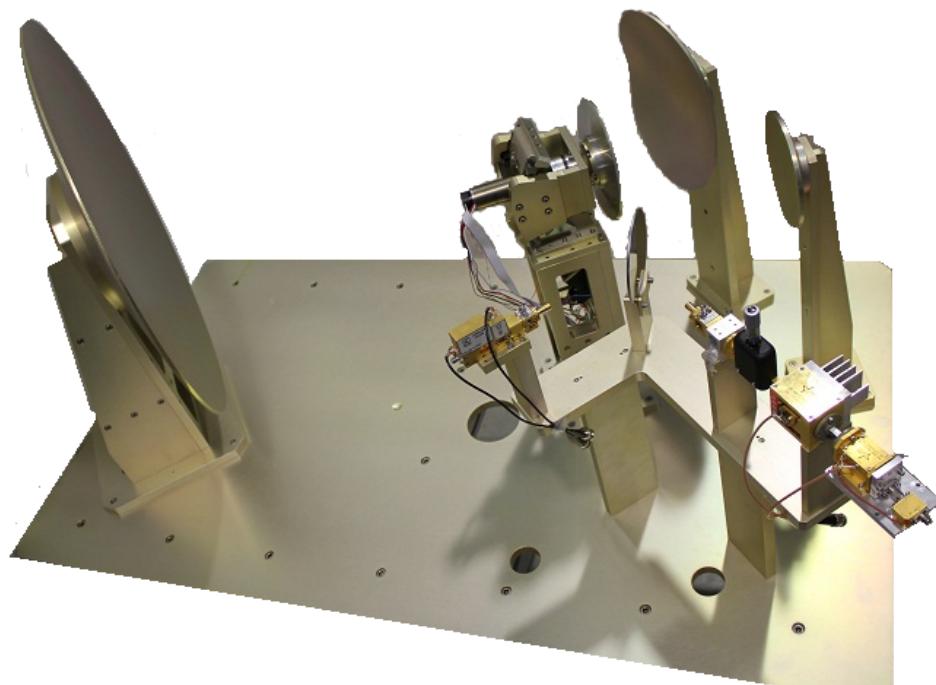


Figura 1.6: Escáner corporal diseñado.

Un dispositivo de similares prestaciones es el desarrollado por el JPL (Jet Propulsion Laboratory), perteneciente a la NASA (National Aeronautics and Space Administration). En la tabla 1.7 se muestra una comparativa entre ambos sistemas.

	JPL	GMR	
Frequency	675	300	(GHz)
Bandwidth	28.8	27	(GHz)
T chirp	0.1	0.0001 - 1744	(ms)
Tx power	-1.5	0	(dBm)
Acquisition time	1	0.5 - 1	(s)
Standoff range	25	8/10	(m)
Spot diameter	1.3	1.3	(cm)
Main antenna aperture	1	0.65	(m)
Field of View	0.4x0.4	0.9x0.5	(m ²)

Figura 1.7: Comparativa entre sistemas.

1.4.2. Sistemas que componen el escáner.

El escáner corporal se estructura en 5 grandes sistemas:

- Sistema de alimentación.
- Sistema de radiofrecuencia.
- Sistema de antenas.
- Sistema de control y sincronismo.
- Sistema de procesado.

En la figura 1.8 se muestra la relación existente entre ellos. Como se puede observar en dicha figura, el sistema de alimentación proporciona energía a los sistemas de radiofrecuencia, de antenas y al sistema de control y sincronismo.

El sistema de radiofrecuencia genera la señal que será radiada posteriormente por el sistema de antenas. Este sistema de antenas dispone a su vez de un subsistema de apuntamiento que permite explorar la zona de interés. El sistema de control y sincronismo se encarga de generar las distintas señales de reloj requeridas por el equipo, de gestionar el subsistema de apuntamiento y de digitalizar la señal recibida, de forma que sea posible realizar un asociación unívoca entre las muestras digitalizadas de la señal de radiofrecuencia recibida y las posiciones de apuntamiento del sistema de antenas. El subsistema de control interacciona con el usuario y configura el subsistema de sincronismo. Por último, el sistema de procesado analiza las muestras facilitadas por el subsistema de digitalización y genera la imagen radar.

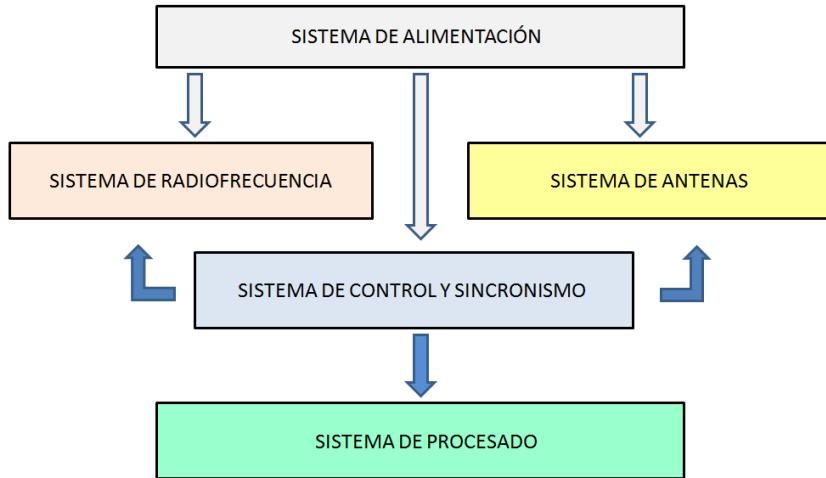


Figura 1.8: Sistemas del escáner corporal.

1.5. Alcance y descripción del proyecto.

Este proyecto tiene por cometido diseñar y fabricar un sistema de control y sincronismo que convierta al escáner en un equipo operativo y muestre imágenes de la escena capturada. También contempla la fabricación de un sistema de alimentación para los distintos componentes. El diseño de los circuitos de microondas y ondas milimétricas que permiten elevar la señal generada hasta los 300 GHz no forma parte de este proyecto. Puesto que el sistema de control y sincronismo interacciona con el resto de sistemas que conforman el radar, se ha decidido describir cada uno de ellos por separado para que sea más fácil su comprensión. No obstante, en el documento se hacen continuas referencias cruzadas entre los distintos sistemas.

El orden elegido para la exposición del contenido de este documento se rige por un criterio que persigue ir describiendo los sistemas de menor a mayor nivel de interacción con el sistema de control y sincronismo. En primer lugar, se describe en el capítulo 2 el sistema de alimentación que, si bien puede interactuar con el subsistema de control, no es necesario que lo haga ya que éste puede operar de forma autónoma. Este proyecto abarca el diseño y fabricación de dicho sistema de alimentación. En realidad se han fabricado dos sistemas de polarización. El primero se ha quedado limitado en capacidad y ha sido necesario desarrollar un segundo sistema que presenta múltiples ventajas frente al primero. En la actualidad este segundo sistema, aunque ha sido diseñado y fabricado, está todavía en fase de pruebas y, por lo tanto, no se ha incorporado al equipo.

Posteriormente, en el capítulo 3 se describe el sistema de antenas. El diseño de dicho sistema no forma parte del alcance del proyecto; sin embargo, este proyecto sí que contempla el análisis del funcionamiento del subsistema de apuntamiento, de su caracterización mediante simulaciones, de medidas realizadas del mismo y de su configuración por

parte del subsistema de control. También se ha desarrollado un sistema de referencias requerido por el subsistema de apuntamiento y otro sistema de referencias necesario para poder posicionar el objeto a explorar en la zona de muestreo del subsistema de apuntamiento.

En el capítulo 4 se detalla el proceso de generación de la señal *chirp* requerida por el equipo. Dicha señal puede ser generada por un DDS (Direct Digital Synthesizer) de la marca Euvis o por otro de la marca Analog Devices. En la actualidad, el equipo funciona con el dispositivo de Euvis, si bien, no se descarta una posible futura implementación del equipo de Analog Devices que, a pesar de ser un equipo operativo, todavía no está integrado en el radar. El proyecto no contempla la configuración y manejo de estos dispositivos aunque sí que han tenido que ser estudiados para determinar la mejor estrategia de sincronismo.

En el capítulo 5 se describe el sistema de control y sincronismo. Este sistema se divide en los subsistemas de sincronismo y de control. En un proceso de exploración, los motores del subsistema de apuntamiento han de sincronizarse con la tarjeta de digitalización y, según el dispositivo utilizado, con el sistema de generación de señal. De esta forma, el sistema de procesado es capaz posteriormente de asociar las muestras de la señal capturada con determinadas posiciones de apuntamiento, logrando de esta manera sintetizar la imagen radar. Tales tareas son realizadas por el subsistema de sincronismo. El subsistema de sincronismo es gobernado por el sistema de control que, además, ofrece una interfaz gráfica que permite al operador del radar su configuración y manejo. Las muestras digitalizadas son almacenadas en un archivo por el subsistema de control. Dicho archivo es posteriormente tratado en Matlab por el sistema de procesado.

El procesado de la señal recibida tampoco pertenece a este proyecto. Este sistema de procesado genera una matriz en la cual se registra el número de muestra asociada a cada dirección de apuntamiento donde se ha determinado la existencia de un blanco, su distancia y su nivel de potencia. No obstante, parte del proyecto incluye la elaboración del código de procesado que permite asociar las muestras digitalizadas en las cuales se ha determinado la existencia de un blanco con la posición del mismo en la escena y del código requerido para la representación de la imagen obtenida (véase el apéndice F.2). Por último, en el capítulo 6 se presentan las posibles mejoras que se proponen y en el capítulo 7 las conclusiones del proyecto.

Capítulo 2

Sistema de alimentación.

2.1. Introducción.

Los distintos componentes que componen el radar necesitan de una alimentación para su funcionamiento. Muchos de ellos permiten un amplio rango de tensiones de entrada, sin embargo, otros requieren de una tensión específica. Es un objetivo de este proyecto diseñar y fabricar una fuente de alimentación que satisfaga las necesidades energéticas del radar.

Las tensiones requeridas por el prototipo inicial del radar y sus potencias se detallan en la tabla 2.1.

TENSIÓN (V)	POTENCIA (W)
24	8
15	4.5
12	21
8	1.2
7	3.15
5	4
-7	8

Tabla 2.1: Tensiones de la fuente de alimentación.

A la hora de realizar un diseño de una fuente de alimentación, se ha de decidir la tipología de fuente: lineal o conmutada. Las fuentes lineales tienen un rendimiento próximo al 40 %; Las conmutadas pueden alcanzar rendimientos superiores al 80 %. Además de suponer un ahorro energético disipan menos calor. Por contra, las fuentes conmutadas presentan un mayor ruido a su salida debido, principalmente, al fenómeno de conmutación. La frecuencia de conmutación de la fuente depende de su diseño. Frecuencias de conmutación elevadas permiten diseños más compactos ya que requieren de capacidades

menores. Las frecuencias típicas de conmutación de este tipo de fuentes oscilan entre las decenas y centenas de kiloHercios. Algunas fuentes de última generación alcanzan frecuencias de conmutación superiores al MegaHercio. Para evitar que el ruido generado por este tipo de fuentes se induzca en algún circuito y pueda ser confundido con blancos de la escena que se analiza, se ha optado por un diseño de fuente lineal.

Los parámetros de diseño de la fuente se han determinado a partir de las fuentes utilizadas en los laboratorios del GMR y con las cuales se han polarizado previamente los componentes del radar hasta la fabricación de esta fuente. La fuente en cuestión es de la marca Agilent y modelo E3631A. Su tensión de rizado es inferior a los 25 mV pico-pico para una tensión de salida de 5V y una resistencia de carga nula. En la figura 2.1 se representa en color marrón la tensión de rizado y en color morado su transformada de Fourier.

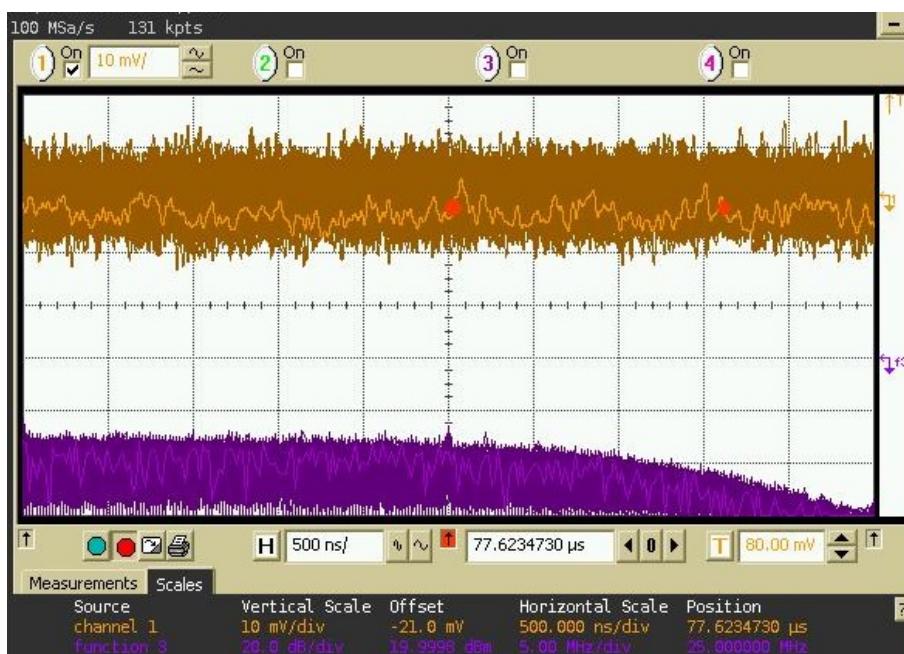


Figura 2.1: Rizado de la fuente de alimentación Agilent E3631A a 5V sin carga.

2.2. Diseño de fuente inicial.

Tomando como referencia esquemas eléctricos de una fuente diseñada previamente por el GMR, se ha fabricado una fuente lineal que suministra las tensiones requeridas. La fuente se integra en una caja de aluminio presofundido de aleación ADC12 de la marca RS y modelo 700-4100 cuyas dimensiones externas son 274 x 173 x 100 mm. La fuente se ha fabricado en dos PCBs distintas para conseguir un diseño más compacto (Véase la figura 2.2). En la superior (mostrada en la parte izquierda de la figura) se

ubican los circuitos de rectificación de cada uno de los devanados secundarios y en la inferior los circuitos de regulación y filtrado.



Figura 2.2: Fuente de alimentación inicial.

La potencia disipada por un regulador lineal se puede aproximar por el producto determinado por la diferencia de tensión entre su entrada y su salida y por la corriente de salida. Para tratar de optimizar el rendimiento de la fuente y minimizar la potencia disipada, se ha procurado que la tensión de entrada de cada regulador sea lo más parecida a la tensión de salida de éste, garantizando en todo momento que la diferencia de tensiones entre la entrada y la salida sea superior a la tensión V_{DROP} requerida por el regulador. Por ello, se ha optado por realizar 4 etapas distintas de rectificación mediante 3 transformadores toroidales según se muestra en la figura 2.3.

La tabla 2.2 detalla las características de cada uno de los tres transformadores. Todos ellos poseen un devanado primario cuya tensión nominal es de 230V. Todos los valores indicados en la tabla se refieren a valores eficaces.

Trafo	Referencia RS	Potencia nominal	Devanados secundarios	Regulación
A	223-7901	50 VA	2x15 V (A1+A2)	14 %
B	540-5163	30 VA	2x9 V (B1+B2)	19 %
C	540-4952	30 VA	2x25 V (C1+C2)	15 %

Tabla 2.2: Transformadores toroidales.

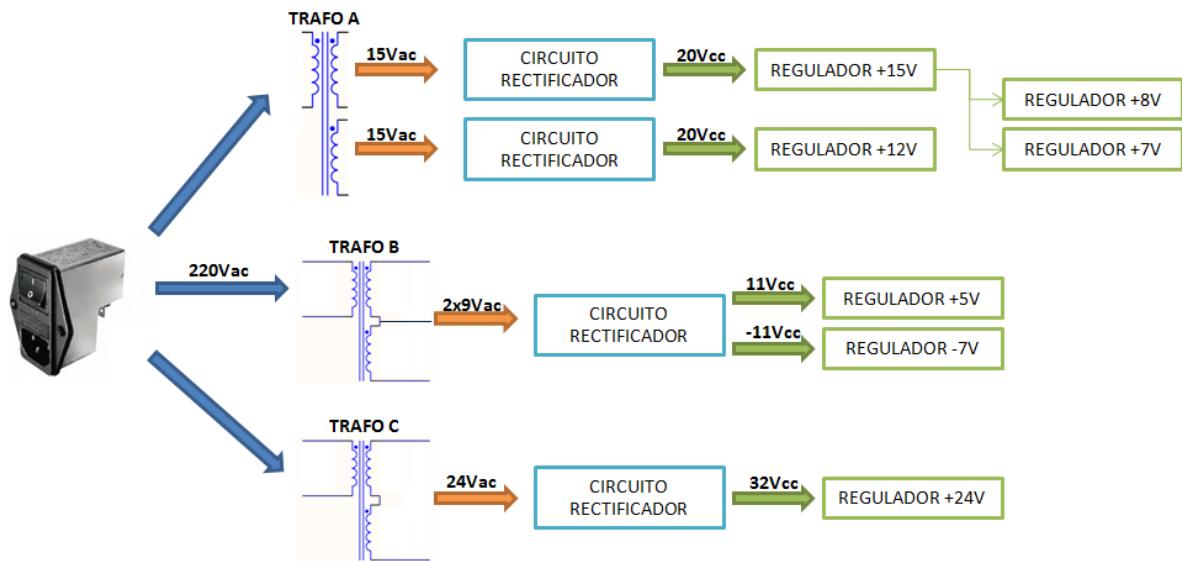


Figura 2.3: Esquema de la fuente.

La fuente dispone de un conector tipo IEC que integra un filtro de red, de un commutador y de un fusible. Por dicho conector la fuente recibe la tensión de red. La señal de red se conecta a los 3 transformadores toroidales en paralelo. El devanado secundario de cada transformador es rectificado mediante un circuito de rectificación de onda completa en la tarjeta de rectificación.

Los reguladores de +8 V y +7 V se alimentan a través del regulador de +15 V para evitar que ambos reguladores disipen potencias elevadas. El rendimiento de la fuente con esta configuración empeora (aumenta 2.8 W el consumo para la misma potencia entregada a la carga) pero las potencias disipadas están más equilibradas entre los 3 reguladores que pasan de disipar 1.8, 5.9 y 1.6 W respectivamente a 1.1, 3.6 y 3.3 W).

En las tablas 2.3, 2.4 y 2.5 se muestra qué bobinado de cada transformador alimenta cada uno de los reguladores, la potencia requerida en la carga por cada tensión de salida y la potencia disipada en cada regulador.

Devanado	Regulador	Potencia entregada	Potencia disipada	Potencia total
A1	15 V	8.9 W	3.3 W	
A1	8 V	1.2 W	1.1 W	
A1	7 V	3.2 W	3.6 W	
A1				16.6 W
A2	12 V	21 W	14.4 W	
A2				35.4 W
A1+A2				52 W

Tabla 2.3: Transformador A.

Devanado	Regulador	Potencia entregada	Potencia disipada	Potencia total
B1	5 V	4 W	5.6 W	
B1				9.6 W
B2	-7 V	8 W	5.8 W	
B2				13.8 W
B1+B2				23.4 W

Tabla 2.4: Transformador B.

Devanado	Regulador	Potencia entregada	Potencia disipada	Potencia total
C1	24 V	8 W	2.6 W	
C1				10.6 W

Tabla 2.5: Transformador C.

2.2.1. Tarjeta de rectificación.

El transformador B conecta sus devanados secundarios en serie referenciando el punto intermedio de unión de ambos (pin 4 del conector P2) a masa según se muestra en la figura 2.4. De esta manera, se consigue obtener una tensión de rectificado negativa.

Hay que tener en cuenta en el diseño de la etapa de rectificación el valor de la capacidad de los condensadores ubicados tras el puente de diodos. Cuanto mayor es el valor de dicha capacidad menor es la tensión de rizado ya que se produce una menor caída de tensión durante en tiempo en el cual los diodos no conducen. Los diodos aportan la corriente demandada por la carga durante el tiempo de conducción (aquellos instantes en los que la tensión del devanado secundario es superior a la suma de las tensiones compuesta por la tensión de los condensadores y la suma de tensiones directas de pico de la pareja de diodos que conducen simultáneamente). Cuanto menor es el tiempo de conducción mayor es la corriente que ha de atravesar la pareja de diodos para suministrar la carga demandada. De tal forma, la capacidad deberá ser lo suficientemente elevada como para mantener una tensión de rectificación con una tensión de rizado razonable y

2. Sistema de alimentación.

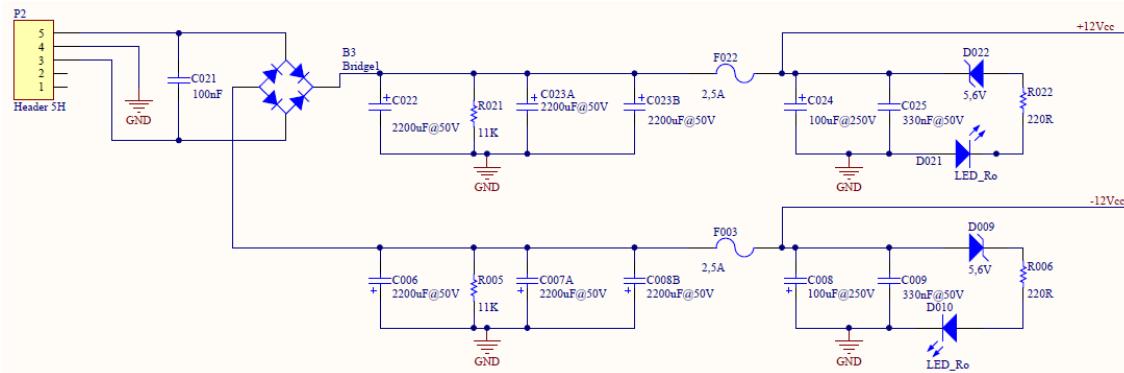


Figura 2.4: Esquema de rectificado del transformador B.

lo suficientemente pequeña como para no provocar una corriente excesiva en el puente de diodos que pueda dañar este último.

En la figura 2.5 se representa en color rojo la tensión del devanado secundario, en color azul la tensión de rectificado y en color verde la corriente que atraviesa una de las parejas de diodos que componen el puente.

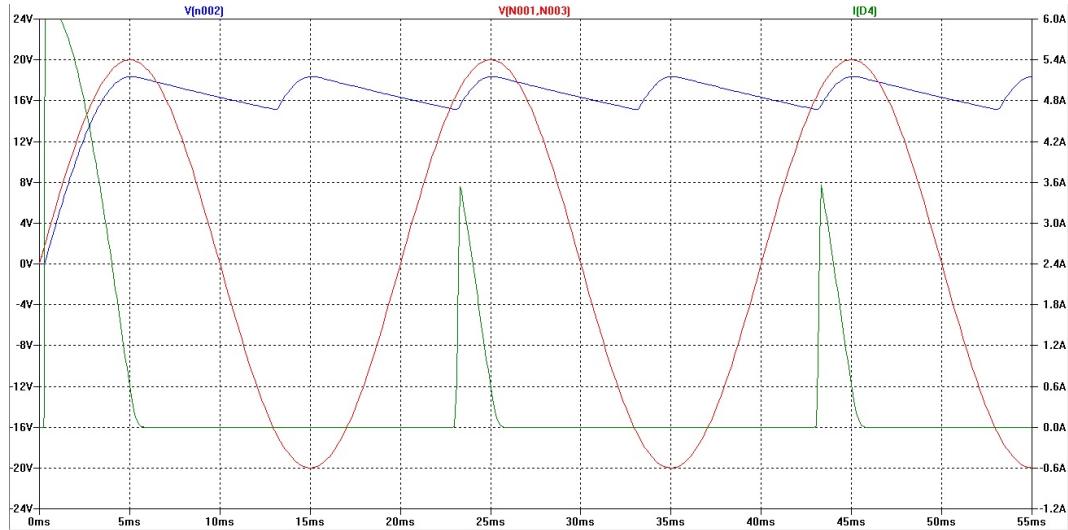


Figura 2.5: Simulación del circuito de rectificación.

El puente de diodos utilizado es de la marca Vishay y modelo GBU4B-E3/51. Soporta una corriente de pico de 150 A y una corriente media de 4 A. A la salida del puente de diodos se conectan 3 condensadores electrolíticos en paralelo que en conjunto presentan una capacidad de 6,600 μ F.

Cada etapa rectificada está dotada de un led que indica la existencia de tensión a su salida y de un fusible de protección de 5 mm. La tarjeta de rectificación transfiere las tensiones rectificadas a la tarjeta de regulación y filtrado.

2.2.2. Tarjeta de regulación.

En la tarjeta de regulación hay una estructura hardware similar por cada tensión a generar. Dicha estructura consta de una célula en π (condensador-bobina-condensador) de filtrado a la entrada, un regulador y otras dos células en π a la salida del regulador. Por último, cada tensión generada dispone a su salida de un fusible de protección.

Las tensiones para las cuales existe un regulador de tensión fijo se han implementado mediante este tipo de reguladores ya que presentan mejores prestaciones que los variables en cuanto a regulación de línea, de carga y de tensión de rizado. Éste es el caso de las tensiones de 24, 15, 12, 8 y 5 V.

Para aquellas tensiones para las cuales no existe versión comercial de dicho regulador (-7 y 7 V) se ha optado por un diseño basado en un regulador ajustable en tensión mediante un potenciómetro multivuelta.

La fuente entrega sus tensiones sobre la tapa de la caja que integra la fuente mediante conectores hembra de tipo banana de 4mm. Los puentes de diodos, los reguladores y los transformadores son los componentes que más potencia disipan. Estos componentes han sido adheridos a las paredes interiores de la caja metálica para facilitar la evacuación del calor que generan. Se han acoplado a la caja disipadores térmicos de gran tamaño para evitar refrigerar la caja mediante el uso de ventiladores. En la figura 2.6 se muestra la fuente fabricada con la que opera actualmente el radar.



Figura 2.6: Fuente lineal.

2.2.3. Caracterización de la fuente.

La figura 2.7 muestra la tensión de salida del circuito rectificador correspondiente al devanado A1 que alimenta al regulador de 7 V. Éste último tiene conectado una

resistencia de carga de 6.8Ω que impone una corriente de salida del regulador de 1 amperio. Se puede observar cómo la tensión de salida del circuito rectificador varía solo 300 mV durante el tiempo de descarga de los condensadores de salida del circuito rectificador.

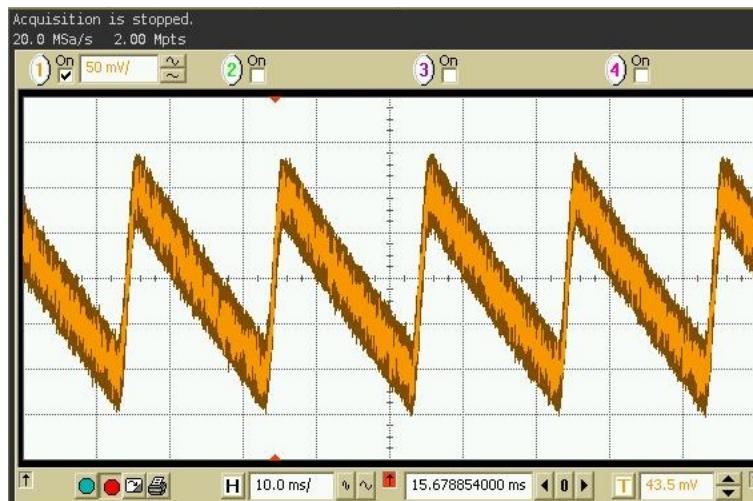


Figura 2.7: Tensión de salida del circuito rectificador del devanado A1.

En la figura 2.8 se muestra la tensión de rizado existente a la salida del regulador de 7 V cuando éste último entrega una potencia de 7.2 W sobre una resistencia de carga de 6.8Ω . La figura 2.9 muestra la misma tensión de rizado para el caso del regulador de 5V con la misma resistencia de carga acoplada a su salida.

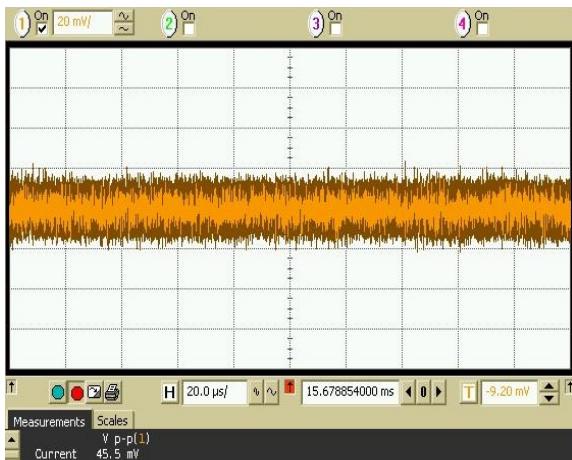


Figura 2.8: Medida tensión de rizado 7 V

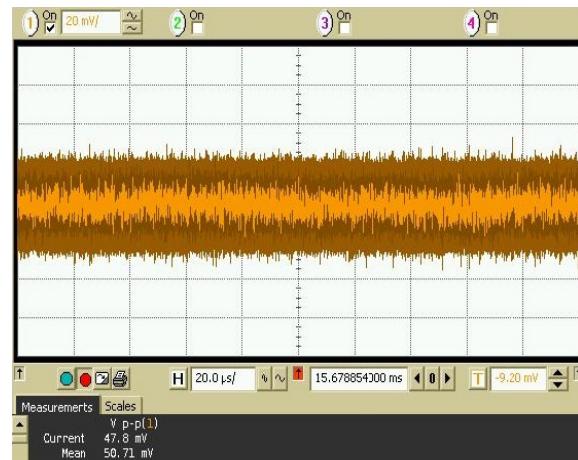


Figura 2.9: Medida tensión de rizado 5 V

2.2.4. Lecciones aprendidas.

Una vez construida e integrada la fuente de alimentación en el radar se han observado errores en su diseño. Se han utilizado unas bobinas en la etapa de filtrado que, si bien tienen un valor adecuado de inductancia para un correcto filtrado en frecuencia, presentan una resistencia serie demasiado alta. Ello provoca una caída de tensión entre los terminales de la bobina del orden de cientos de mV que depende de la corriente demandada por la carga. Dicho error podría haber sido detectado y corregido en una fase inicial del diseño si el comportamiento de ésta hubiese sido simulado mediante software.

Otro factor que se deberá tener en cuenta en futuros diseños es la facilidad de reparación y mantenimiento del equipo. En la fuente actual, la detección de una avería o reparación de la misma requiere de tediosas labores de montaje y desmontaje tales como desacoplar reguladores y puentes de diodos adheridos a la pared interior de la caja. Además, la fuente no puede suministrar nuevas tensiones sin un rediseño completo de la misma. Para subsanar estos inconvenientes se ha de optar en el futuro por un diseño modular, ampliable y de fácil mantenimiento.

Por otro lado, el diseño de la fuente se ha concebido incorporando todos los elementos que la componen dentro de una caja metálica sin aperturas. El objetivo de tal estrategia es evitar que señales de radiofrecuencia externas se puedan inducir en los circuitos de la fuente y afecten a su correcto funcionamiento. Al no existir una convección adecuada de aire, es necesario emplear disipadores de grandes dimensiones acoplados en el exterior de la caja que eviten que el calor se acumule en el interior de la caja. Es recomendable en futuros diseños utilizar cajas metálicas que ofrezcan un aislamiento electromagnético adecuado y a su vez permitan un flujo de aire que facilite la refrigeración de los componentes.

Por último, la fuente diseñada no facilita información de la tensión y corriente entregada a cada carga ni dispone de circuitos de limitación de corriente que eviten posibles daños a los componentes. En un sistema complejo como puede ser el radar en cuestión, es difícil identificar el origen de una avería que, en muchas ocasiones, se debe a un simple fusible fundido. Por ello se recomienda que en futuras versiones la fuente pueda tener un sistema de monitorización y un sistema de limitación de corriente.

2.3. Diseño de fuente modular.

Se ha de realizar el diseño de una segunda versión de la fuente de alimentación que integre las mejoras propuestas en el anterior apartado. La nueva fuente debe proporcionar las tensiones detalladas en la tabla 2.6. Las tensiones de 3.3, 1.8 y -5 V no estaban implementadas en la fuente inicial.

TENSIÓN (V)	POTENCIA (W)
24	8
15	4.5
12	21
8	1.2
7	3.15
5	10
3.3	2
1.8	1
-5	12
-7	8

Tabla 2.6: Tensiones de la fuente modular.

El diseño de la nueva fuente (mostrada en la figura 2.10) se basa en un chasis modular preparado para albergar 3 tarjetas de rectificación, 7 tarjetas reguladoras capaces de generar un total de 14 tensiones diferentes y una tarjeta controladora que permite monitorizar y gestionar la fuente a través de un PC mediante una conexión TCP/IP.

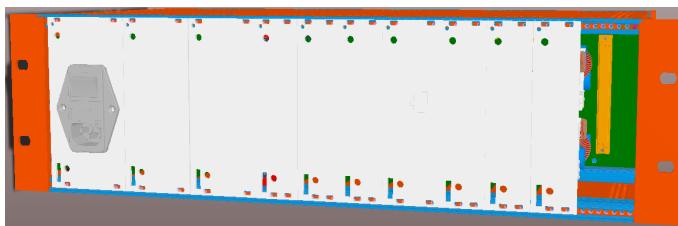


Figura 2.10: Fuente modular.

Dada la gran variedad de tensiones que se han de generar, debería ser necesario incorporar múltiples transformadores con diferentes factores de transformación para evitar disipar potencias elevadas en los reguladores. No obstante, gracias a un nuevo circuito basado en la guía de diseño para reguladores lineales de alta eficiencia del fabricante Linear Technologies [20], un sistema comutado previo al regulador ajusta la tensión de entrada al mismo que minimiza su consumo (véase el apartado 2.3.4). Mediante esta técnica y haciendo uso de reguladores LDO (Low DropOut) se logran eficiencias cercanas al 80% manteniendo un nivel de ruido similar al de las fuentes lineales.

La fuente, mostrada en la figura 2.11, está actualmente en proceso de pruebas y todavía no se encuentra implementada en el radar.

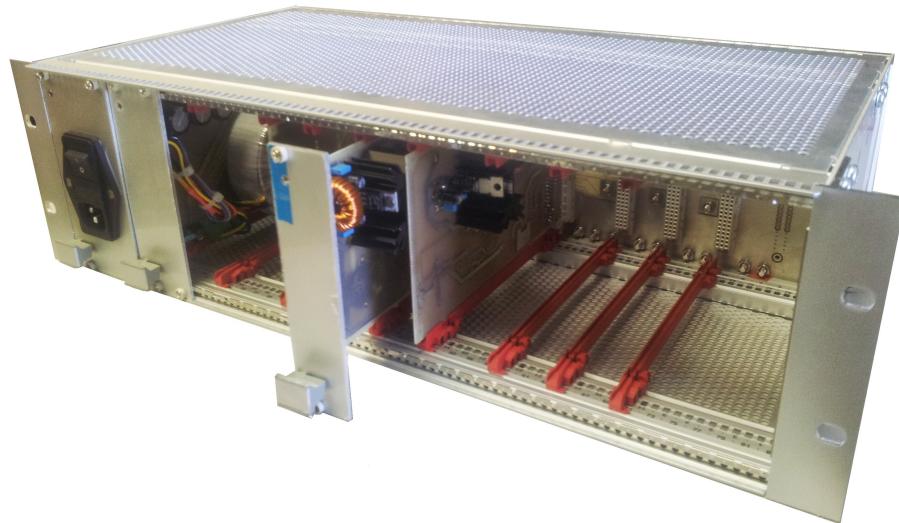


Figura 2.11: Fuente modular.

2.3.1. Rack.

Las distintas tarjetas que componen la fuente se insertan en un rack de 19" de aluminio de la marca SCHROFF y modelo 24563-152 mostrado en la figura 2.12.

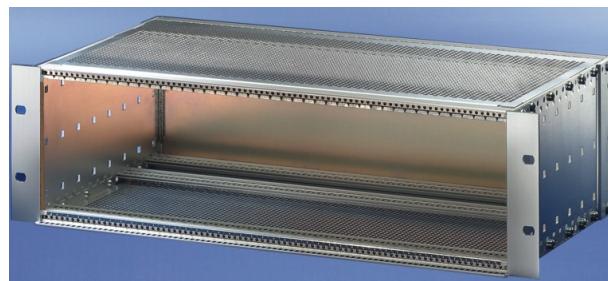


Figura 2.12: Rack de la fuente modular.

Dicho rack tiene una altura estandarizada de 3 U's y dispone de un dimensionamiento horizontal para albergar tarjetas de hasta 84 HP's (unidades estandarizadas).

El rack acepta tarjetas de tipo Eurocard VME de 3U de altura y 160 mm de profundidad. Dispone de perforaciones en todas sus paredes que permiten la circulación de flujo de aire para su ventilación. A pesar de dichas perforaciones, el rack garantiza un blindaje electromagnético de 40 dB a 1 GHz y de 30 dB a 2 GHz.

2.3.2. Backplane.

Las distintas tarjetas que componen la fuente se interconexinan mediante un *backplane* ubicado en la parte posterior del rack según se muestra en la figura 2.13.

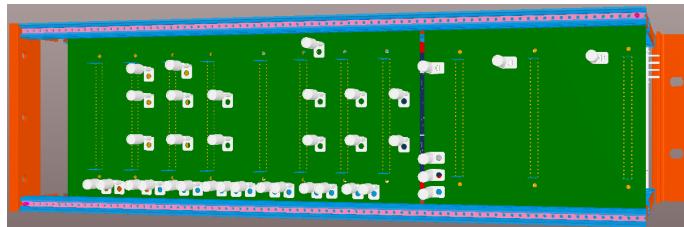


Figura 2.13: Rack de la fuente modular.

Las tarjetas se insertan en el *backplane* mediante los conectores DIN 41612 hembra de 64 vías que incorpora éste en su cara anterior. En su cara posterior incorpora conectores de tipo *faston* de 6.35 mm para facilitar la conexión de los distintos dispositivos a la fuente. El backplane proporciona 11 slots para tarjetas según muestra el esquema de la figura 2.14

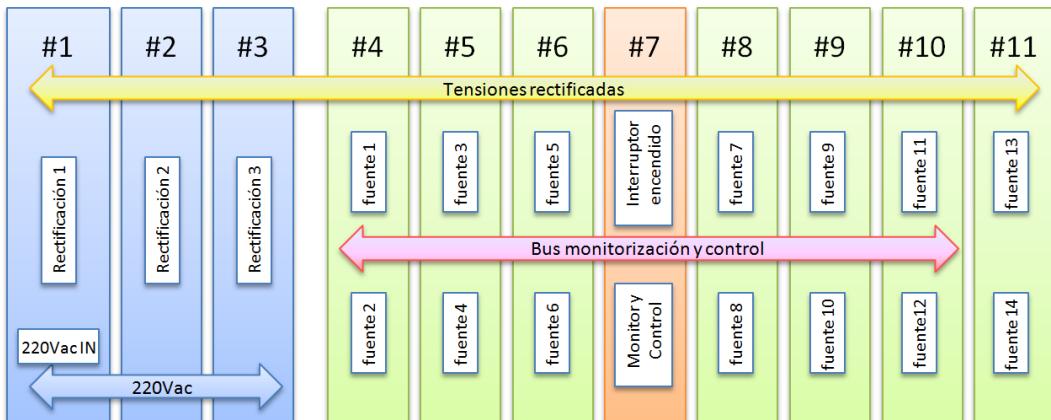


Figura 2.14: Estructura del *Backplane*.

El *backplane* ha tenido que ser diseñado en 2 PCBs por limitaciones en el proceso de fabricación. Las cubetas donde se sumergen los circuitos en el ácido que ataca la superficie de cobre tienen unas dimensiones limitadas. Las tarjetas, al igual que el resto de tarjetas de la fuente, están fabricadas con un sustrato *FR4* (*Flame Retardant 4*) de 1.6 milímetros de espesor. Ambas caras de dicho dieléctrico están recubiertas por una capa de cobre de 35 μm .

Para evitar en la medida de lo posible caídas de tensión en las interconexiones entre placas del *backplane*, se ha decidido dividir éste separando las tarjetas de rectificación de las de regulación. De esta manera, la primera tarjeta *backplane* mostrada en la figura 2.15, integra las 3 tarjetas de rectificación.

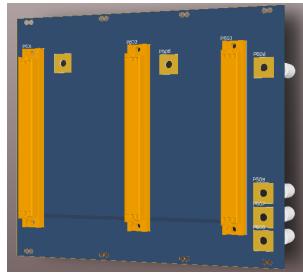


Figura 2.15: *Backplane* de rectificación.

Esta tarjeta replica la tensión de red de 220 V introducida a través de la primera tarjeta de rectificación a las otras dos. En la segunda tarjeta (figura 2.16) se acoplan las tarjetas de regulación y la tarjeta controladora. Dicha tarjeta controladora ha de comunicarse con todas las tarjetas de regulación. Por este motivo se ha decidido ubicar esta tarjeta en uno de los *slots* centrales de esta segunda placa.

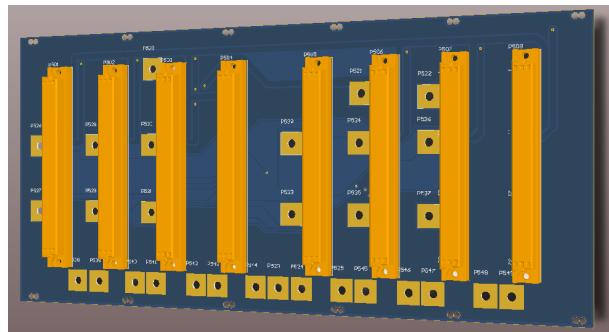


Figura 2.16: *Backplane* de regulación.

Ambas tarjetas *backplane* se interconexinan mediante conectores *faston* iguales a los anteriormente descritos. Para lograr que todos los componentes queden alineados correctamente entre las diferentes tarjetas y el chasis ha sido imprescindible trabajar con modelos 3D de todas las piezas involucradas en el diseño.

2.3.3. Tarjetas de rectificación.

La rectificación de la fuente se realiza en tres tarjetas, dos de ellas rectifican tensiones positivas y la última rectifica una tensión negativa.

La rectificación se ha dividido en varias tarjetas para lograr que los transformadores toroidales que éstas incorporan quepan en el rack.

La primera tarjeta rectificadora, mostrada en la figura 2.17, incorpora un conector IEC de tipo C14 en su carátula frontal por el cual se introduce la alimentación de red a toda la fuente. Dicho conector IEC implementa un interruptor de encendido y apagado, un filtro de red y un fusible de 5mm.

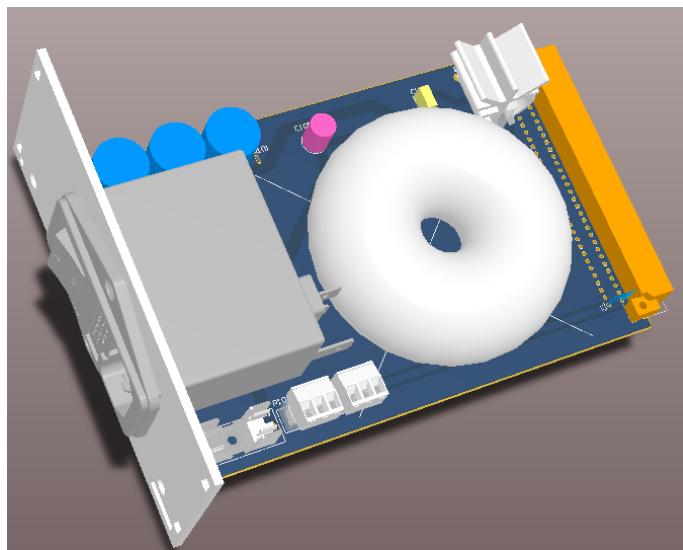


Figura 2.17: Tarjeta de rectificación #1.

Esta primera tarjeta incorpora un transformador toroidal de la marca Nuvotem y modelo 91931-P1S1 de 50 VA. Posee un único devanado secundario que ofrece una tensión eficaz de 11.8 Vac cuando el devanado primario es excitado con 220 Vac.

La rectificación aplicada es de onda completa y se realiza a través de un puente de diodos de la marca Vishay y modelo GBU4B-E3/51. Dicho puente está equipado con un dissipador térmico cuya resistencia es de 7.8 °K/W que se muestra en la parte superior derecha de la figura 2.17.

A la salida de dicho rectificador se ha incorporado un portafusibles para fusibles cilíndricos cuyas dimensiones han de ser 5 x 20 mm. La tarjeta también suministra al *backplane* de rectificación los 220 V de la red eléctrica.

La segunda tarjeta rectificadora es igual a la anterior (a excepción del conector IEC que no esta incorporado). El transformador que integra es de la marca RS y modelo 12022. La potencia aparente o módulo de la impedancia compleja ($|V \cdot I^*|$) del transformador es de 50 VA. Posee un devanado primario de 220 Vac y 2 devanados secundarios de 12 Vac que están conectados en serie, de forma que, proporcionan una tensión eficaz de 24 Vac como se observa en la figura 2.18.

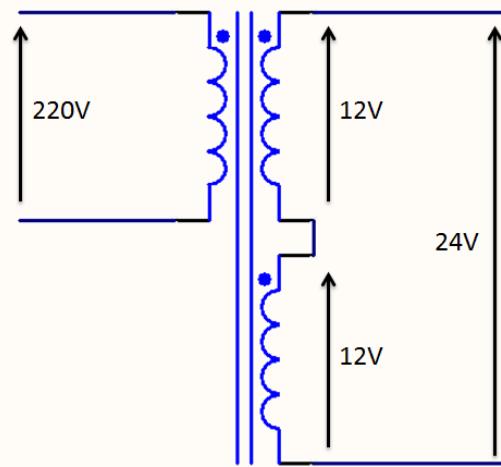


Figura 2.18: Conexiónado en serie de los devanados secundarios.

La tercera tarjeta rectificadora (figura 2.19) es la encargada de rectificar tensiones negativas. Posee un transformador de 15 VA de la marca Nuovotem y modelo 70050K. Dicho transformador posee un devanado primario de 220 Vac y dos secundarios de 7V. Al igual que en el transformador anterior, ambos devanados están conectados en serie.

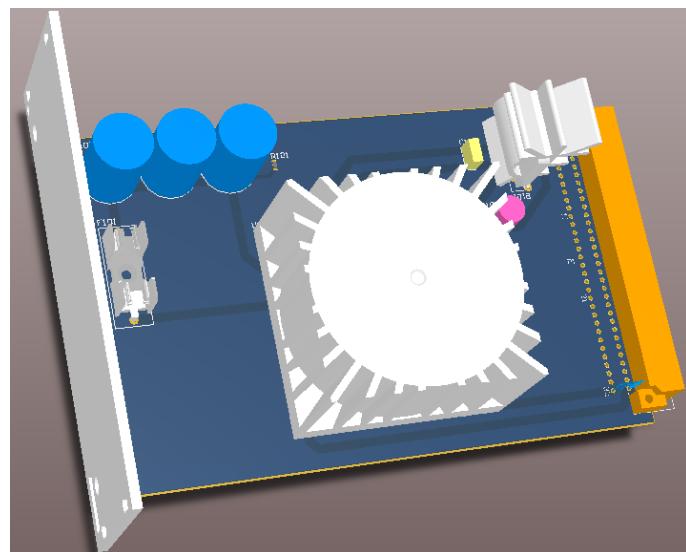


Figura 2.19: Tarjeta de rectificación #3.

2.3.4. Tarjeta de regulación.

La tarjeta de regulación (figura 2.20) recibe las tensiones facilitadas por las tarjetas rectificadoras.

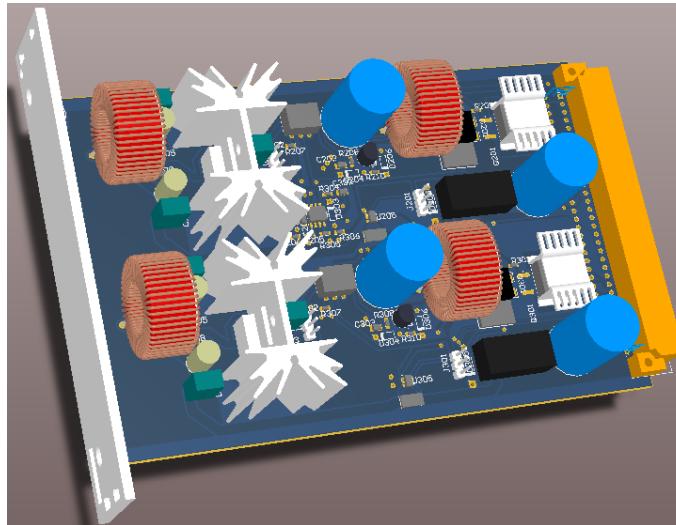


Figura 2.20: Tarjeta de regulación.

Dispone de 2 circuitos reguladores iguales que incorporan un *jumper* a modo de selector y permite determinar qué tensión de rectificado utilizará cada circuito. El circuito de regulación implementa un regulador LDO (Low DropOut) de la marca Linear Technologies y modelo LT1083 capaz de entregar hasta 7.5 amperios con una tensión de salida ajustable entre 1.25 y 30 V. El regulador garantiza una regulación de linea de 0.2 % (definida según la ecuación 2.1) y una regulación de carga de 0.3 % (definida por la ecuación 2.2). Su tensión V_{DROP} (mínima diferencia de tensión entre la entrada y salida para que el regulador trabaje correctamente) es inferior a 1.5 V. Está acoplado a un disipador térmico igual al descrito en el apartado 2.3.3.

$$Reg_{línea} = \frac{\frac{\Delta V_{out}}{V_{out}}}{\Delta V_{in}} \cdot 100 (\%) \quad (2.1)$$

$$Reg_{carga} = \frac{\frac{\Delta V_{out \sin Carga} - V_{out \carga M\acute{a}xima}}{V_{out \carga M\acute{a}xima}}}{\Delta I_{carga}} \cdot 100 (\%) \quad (2.2)$$

El circuito regulador, representado esquemáticamente en la figura 2.21, incorpora un sistema de conmutación previo al regulador que ajusta la tensión de entrada al regulador a un valor igual a su tensión de salida más 1.7 V (V_{DROP} más un margen de 0.2 V de seguridad). Mediante esta técnica se logra que el regulador tenga una tensión de entrada suficiente para poder regular correctamente y se minimiza la potencia disipada en el mismo.

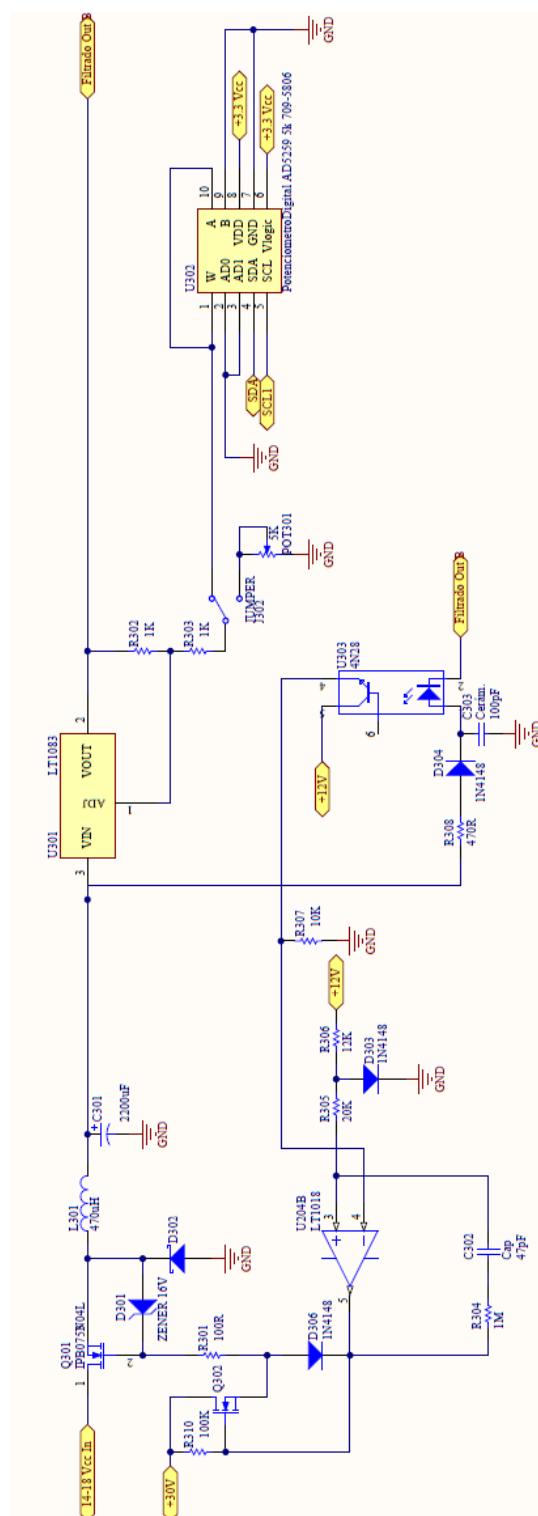


Figura 2.21: Circuito de commutación de la tarjeta de regulación.

La etapa previa de conmutación varía su frecuencia de trabajo en función de la tensión de entrada, de la de salida y de la corriente demandada por la carga. La tensión de entrada al regulador tiene un rizado de aproximadamente 100 mVpp y puede alcanzar una frecuencia de hasta 6 KHz. Dicho rizado es mitigado por el regulador. La figura 2.22 muestra, para 2 frecuencias distintas determinadas por el fabricante, el rechazo al rizado de la señal de entrada en función de la corriente de salida normalizada.

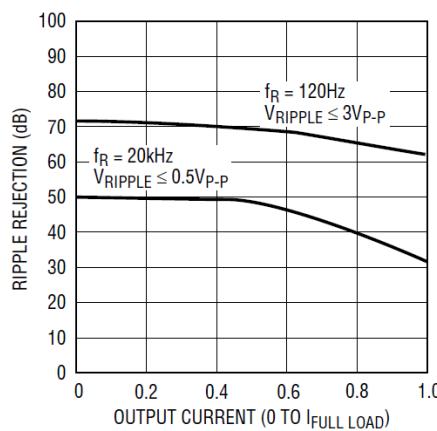


Figura 2.22: Rechazo a la tensión de rizado del regulador LT1083.

Para explicar el funcionamiento del circuito mostrado en 2.21 de una manera más sencilla, se procederá a describir el mismo mediante una primera simplificación a la que posteriormente se le irán añadiendo más componentes hasta llegar al diseño final.

En un primer modelo de simplificación el regulador U201 de la figura 2.23 proporciona una tensión de salida fija. La tensión de entrada al regulador es controlada por el transistor MOSFET Q201 que carga el condensador C201 a través de la bobina L201. El transistor es gobernado por el comparador U204 el cual compara la tensión de entrada al regulador respecto de una referencia de tensión. Cuando la tensión de entrada es inferior al valor de la referencia, el transistor comuta y carga el condensador. La tensión de referencia es igual a la tensión de salida más 1.7 V.

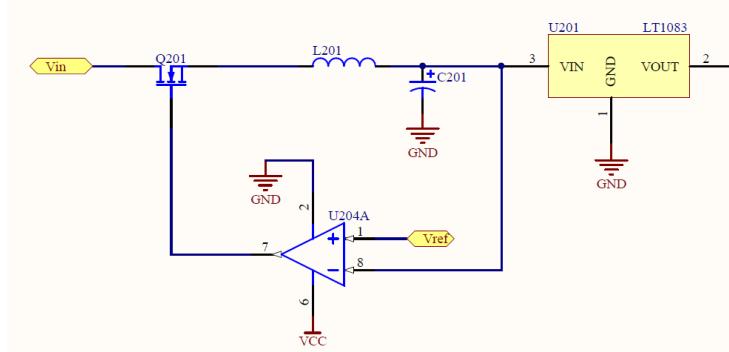


Figura 2.23: Circuito de regulación simplificado I.

En la figura 2.24 se muestra un modelo más aproximado en el cual la tensión de salida del regulador se ajusta mediante el divisor resistivo constituido por R202 y R203.

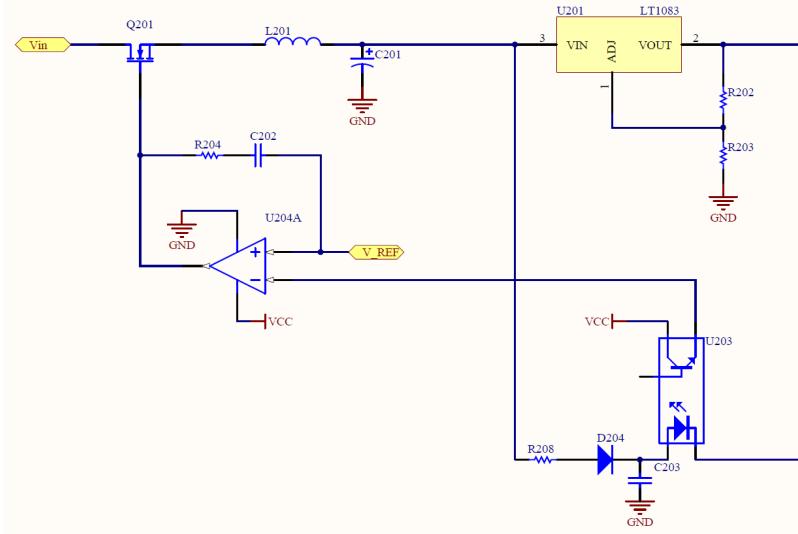


Figura 2.24: Circuito de regulación simplificado II.

En este modelo, el diodo emisor del optoacoplador U203 comienza a conducir cuando la diferencia de tensión entre la entrada y la salida del regulador es superior a 1.6 V. La tensión de salida del optoacoplador continúa aumentando conforme dicha diferencia de tensión aumenta. Cuando la tensión de salida del optoacoplador supera el valor de la referencia, el comparador bascula y corta la carga del condensador C203.

Para reducir la frecuencia de conmutación en el transistor y así disminuir el ruido de alta frecuencia, se introduce una histéresis de 100 mV en el comparador U204 mediante una realimentación positiva formada por R204 y C202. En la figura 2.25, se representa en color rojo la evolución de la tensión de entrada al comparador y en color azul la tensión de salida del mismo. En ella se puede observar el periodo entre conmutaciones del transistor que se corresponde con una frecuencia de 5.3 KHz.

Es necesario que la tensión V_{GS} en Q201 sea lo suficientemente elevada como para garantizar que el transistor trabaje en modo saturado y, por lo tanto, se ha de polarizar la puerta del transistor con una tensión superior a la tensión de drenador. Para ello el comparador excita el transistor Q202 de la figura 2.26.

La tensión V_{GS} máxima soportada por el transistor Q201 es de ± 16 V. Para evitar superar estos límites se han añadido el diodo zener D201 y el diodo Schottky D202. El diodo zener de 16V impide que se supere el umbral positivo y el diodo D202 evita los picos de tensión negativos que se originan en la bobina L201 cuando el transistor interrumpe repentinamente la corriente.

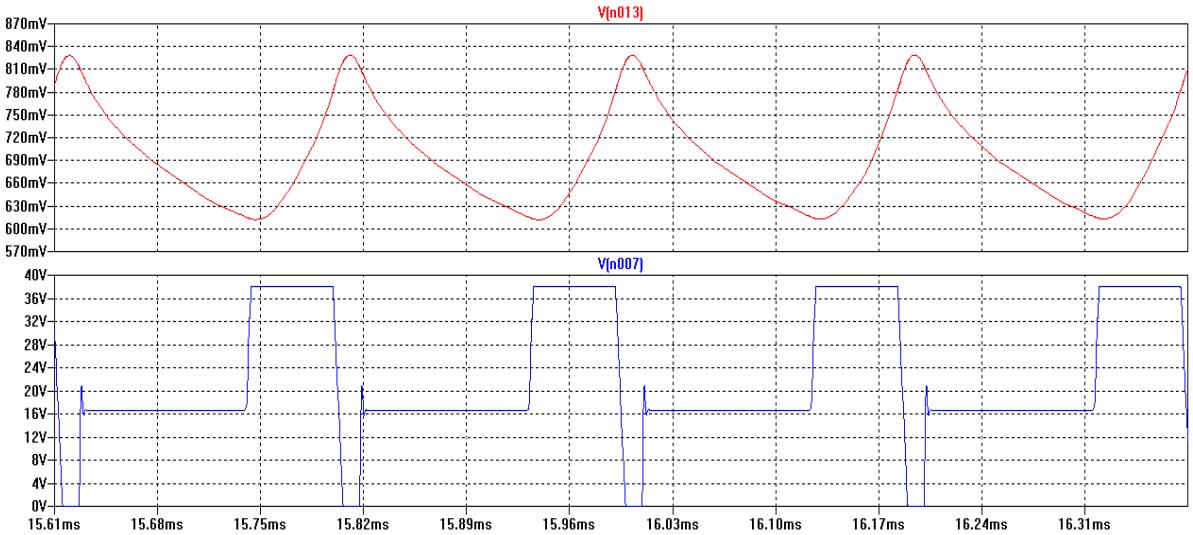


Figura 2.25: Señales de entrada y salida del comparador U204.

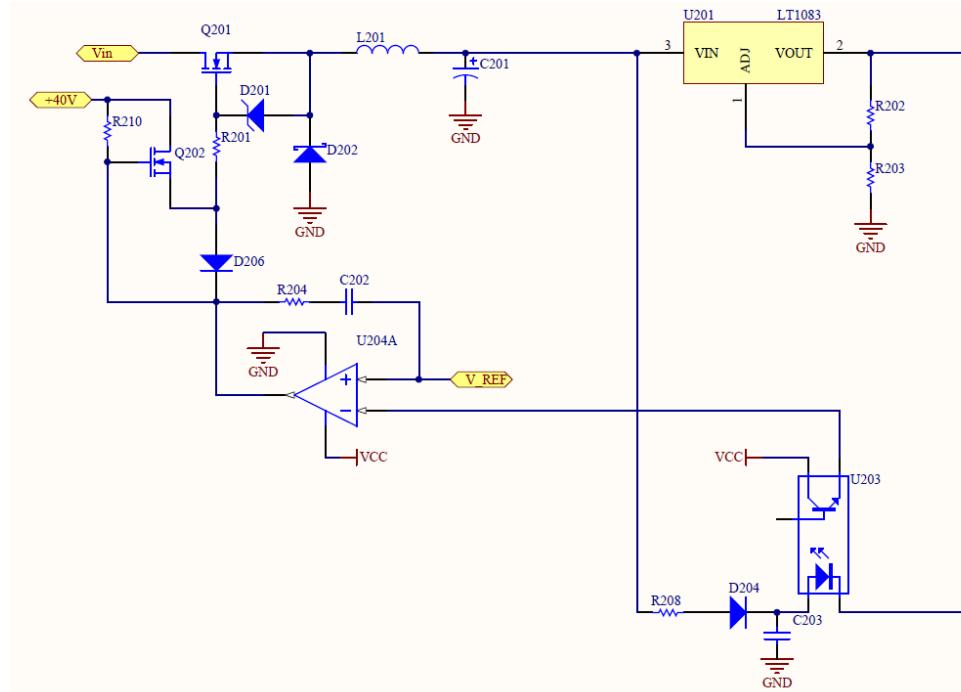


Figura 2.26: Circuito de regulación simplificado IV.

Además del rechazo al rizado que proporciona el regulador, comprendido entre los 30 y 70 dB, se ha implementado a la salida del regulador un filtro paso bajo según se muestra en la figura 2.27 que reduce todavía más el rizado.

El ajuste de la tensión de salida del regulador se realiza mediante hardware o software. Dicha elección depende de la configuración establecida mediante el jumper J302

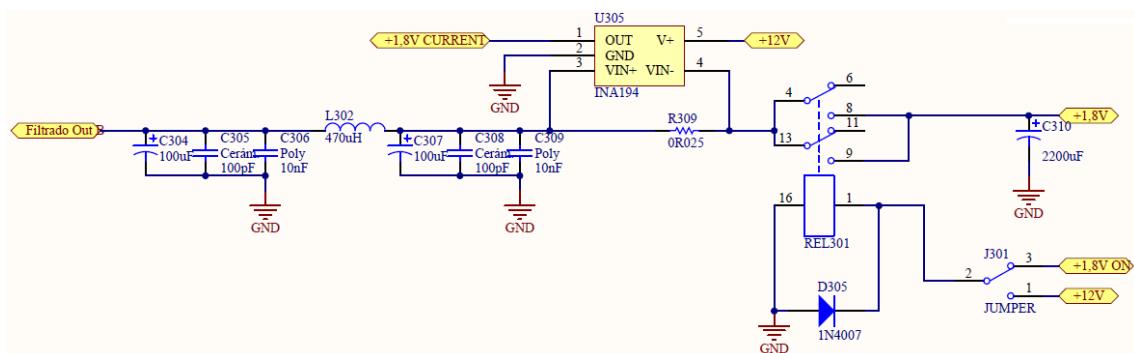


Figura 2.27: Circuito de filtrado de la tarjeta de regulación.

(véase la figura 2.21). En el caso de seleccionarse un ajuste hardware, dicho ajuste se realiza con un potenciómetro multivuelta de valor $5\text{ K}\Omega$ (POT301). Cuando el ajuste es software, un potenciómetro digital (U302) del mismo valor gobernado por la tarjeta controladora (véase el apartado 2.3.5) realiza dicho ajuste.

La tensión de salida de cada fuente permite un rango comprendido entre 1.28V y la tensión máxima entregada por la tarjeta rectificadora menos 1.6 voltios.

La corriente de salida del regulador atraviesa una resistencia de precisión de tolerancia 1% a modo de *shunt* de corriente. El valor de dicha resistencia se elige entre $0.1\ \Omega$ y $0.025\ \Omega$ en función de la tensión de salida de la fuente y de la corriente demandada por la carga para lograr que la tensión de caída entre sus bornes se aproxime a los 50 mV. Dicha tensión es amplificada por el componente U305. De esta manera, el amplificador entrega a su salida una señal en tensión comprendida entre 0 y 12 V que es proporcional a la corriente de salida del regulador.

Dicha tensión amplificada, al igual que la propia tensión del regulador, se refleja en el conector de salida DIN 41612 macho para que la tarjeta controladora pueda monitorizar tanto la tensión como la corriente de salida de cada regulador.

Además, un relé (REL301) es capaz de commutar la tensión de salida. El relé puede estar gobernado mediante hardware (relé permanentemente activo) o mediante software (gobernado por la tarjeta controladora). La selección del modo de operación depende del ajuste del potenciómetro J301.

Se ha realizado una simulación del circuito de regulación mediante el software LT-Spice IV que proporciona Linear Technologies (fabricante de los reguladores y de los comparadores). Dicho software incorpora un modelo detallado del comportamiento de ambos componentes. En la figura 2.28 se muestra el esquema de simulación. En ella se han configurado unos valores de R9 y R10 que determinan una tensión de salida de 24 voltios y se ha añadido la resistencia de carga a la salida R8 que impone una corriente de 2 amperios.

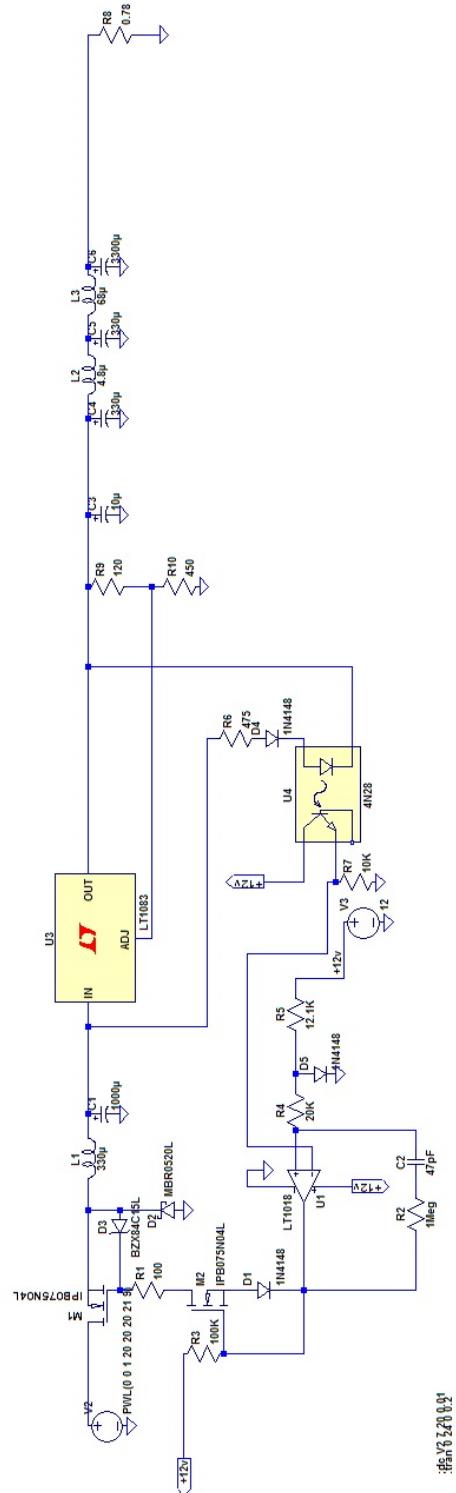


Figura 2.28: Esquema de simulación del circuito de regulación.

En la figura 2.29 se representa en color a rojo la evolución de la tensión de entrada al regulador, en color azul la tensión de salida y en color verde la corriente de salida para una tensión de rectificación de 36V. Se puede apreciar como la tensión de entrada al regulador se ajusta a 1.7 V por encima de la tensión de salida.

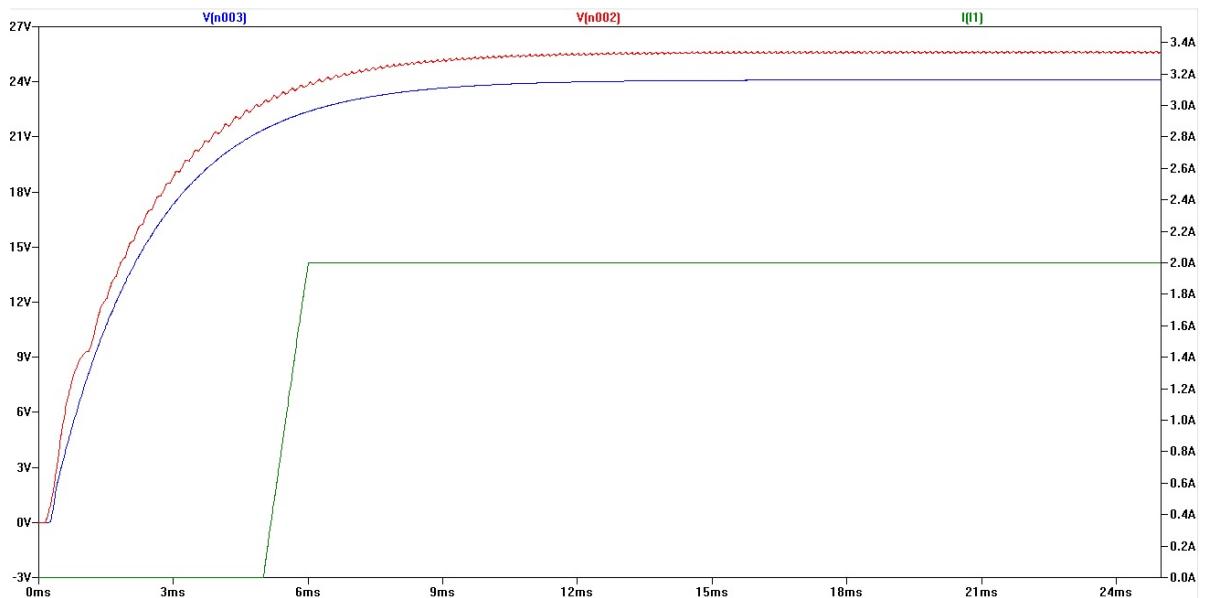


Figura 2.29: Resultado de la simulación del circuito de regulación I.

En la figura 2.30 se muestran los resultados de la simulación para los puntos del circuito definidos en la tabla 2.7.

LEYENDA	MEDIDA
V(n005)	Voltaje en puerta del transistor M1
I(D2)	Corriente que atraviesa el diodo D2
I(L1)	Corriente que atraviesa la bobina L1
V(n013)	Voltaje a la salida del optoacoplador U4
V(n010)	Voltaje en pin positivo del comparador U1
V(N006)*Ix(U3:ADJ)...	Potencia disipada en el regulador U3
V(n002)	Voltaje a la entrada del regulador U3
I(D3)	Corriente que atraviesa el diodo D3

Tabla 2.7: Descripción de las medidas.

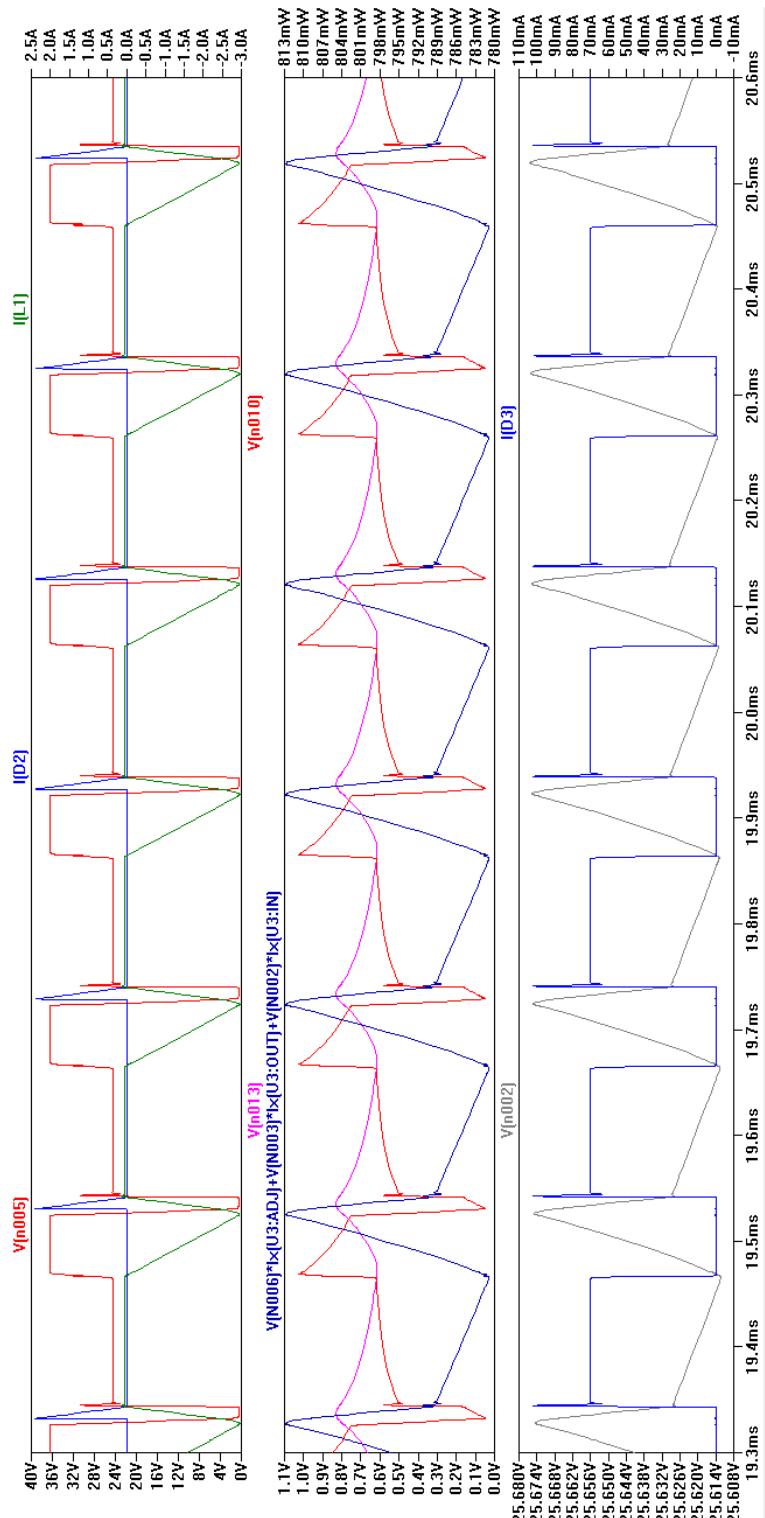


Figura 2.30: Resultado de la simulación del circuito de regulación II.

La tabla 2.8 muestra la eficiencia de cada regulador para todas las tensiones de la fuente. La potencia disipada se ha aproximado por el producto determinado por la diferencia de tensión entre pines de entrada y salida del regulador y por la corriente de salida.

Tensión(V)	Potencia entregada(W)	Potencia disipada(W)	Eficiencia(%)
24	8	0.5	94
15	4.5	0.5	90
12	21	2.8	88
8	1.2	0.2	83
7	3.15	0.7	81
5	10	3.2	76
3.3	2	1.0	67
1.8	1	0.9	53
-5	12	3.8	76
-7	8	1.8	81

Tabla 2.8: Eficiencia de las fuentes.

2.3.4.1. Análisis térmico.

El fabricante de los reguladores facilita un modelo térmico que incluye 2 partes diferenciadas: la sección de control y la sección del transistor de potencia. Cada una de estas secciones tienen un valor de resistencia térmica respecto del encapsulado distinta (Θ_{sc} y Θ_{tp} respectivamente). En la tabla 2.9 se muestran los valores de resistencia y la temperatura máxima soportada para cada una de las uniones.

Unión	Resistencia al encapsulado($^{\circ}\text{C}/\text{K}$)	Temperatura máxima($^{\circ}\text{C}$)
Sección de Control Θ_{sc}	0.5	125
Sección transistor de potencia Θ_{tp}	1.6	150

Tabla 2.9: Modelo térmico del regulador.

Se ha estimado una temperatura interior del rack de la fuente de $T_{amb} = 60 \text{ }^{\circ}\text{C}$ para una convección de aire no forzada según lo recomendado en [22]. Cada regulador está adherido a su disipador mediante una pasta térmica que se modela como una resistencia térmica de $\Theta_p=0.2 \text{ }^{\circ}\text{K}/\text{W}$. La resistencia térmica del disipador es de $\Theta_{dis}=7.8 \text{ }^{\circ}\text{K}/\text{W}$. Se ha estimado la temperatura de ambas uniones (T_{sc} =sección control y T_{tp} =sección transistor de potencia) y del disipador según las fórmulas 2.3, 2.4 y 2.5, donde P_{dis} es la potencia media obtenida en la simulación mostrada en la figura 2.31.

$$T_{sc} = T_{amb} + P_{dis} \cdot (\Theta_{sc} + \Theta_p + \Theta_{dis}) \quad (2.3)$$

$$T_{tp} = T_{amb} + P_{dis} \cdot (\Theta_{tp} + \Theta_p + \Theta_{dis}) \quad (2.4)$$

$$T_{dis} = T_{amb} + P_{dis} \cdot \Theta_{dis} \quad (2.5)$$

Tensión(V)	Corriente(A)	P_{dis} =Potencia media disipada(W)	T_{sc} (°C)	T_{tp} (°C)	T_{dis} (°C)
24	0.5	0.79	67	68	66
12	0.5	0.81	67	68	66
8	0.5	0.82	67	68	66
5	0.5	1.13	70	71	69
3.3	0.5	1.73	75	77	73
24	2	3.02	86	89	84
12	2	3.13	87	90	84
8	2	3.20	87	91	85
5	2	3.56	90	94	88
3.3	2	4.26	96	101	93
24	3.5	5.20	104	110	101
12	3.5	5.43	106	112	102
8	3.5	5.56	107	113	103
5	3.5	5.95	111	117	106
3.3	3.5	6.70	117	124	112

Tabla 2.10: Temperaturas del regulador.

Las potencias demandadas por el sistema (indicadas en la tabla 2.6) permiten que la fuente sea capaz de funcionar sin una convección de aire forzada. Para corrientes superiores a 2 A se recomienda instalar un ventilador. La figura 2.31 muestra los resultados de las simulaciones de la potencia disipada por los reguladores para las tensiones y corrientes indicadas en la tabla 2.10

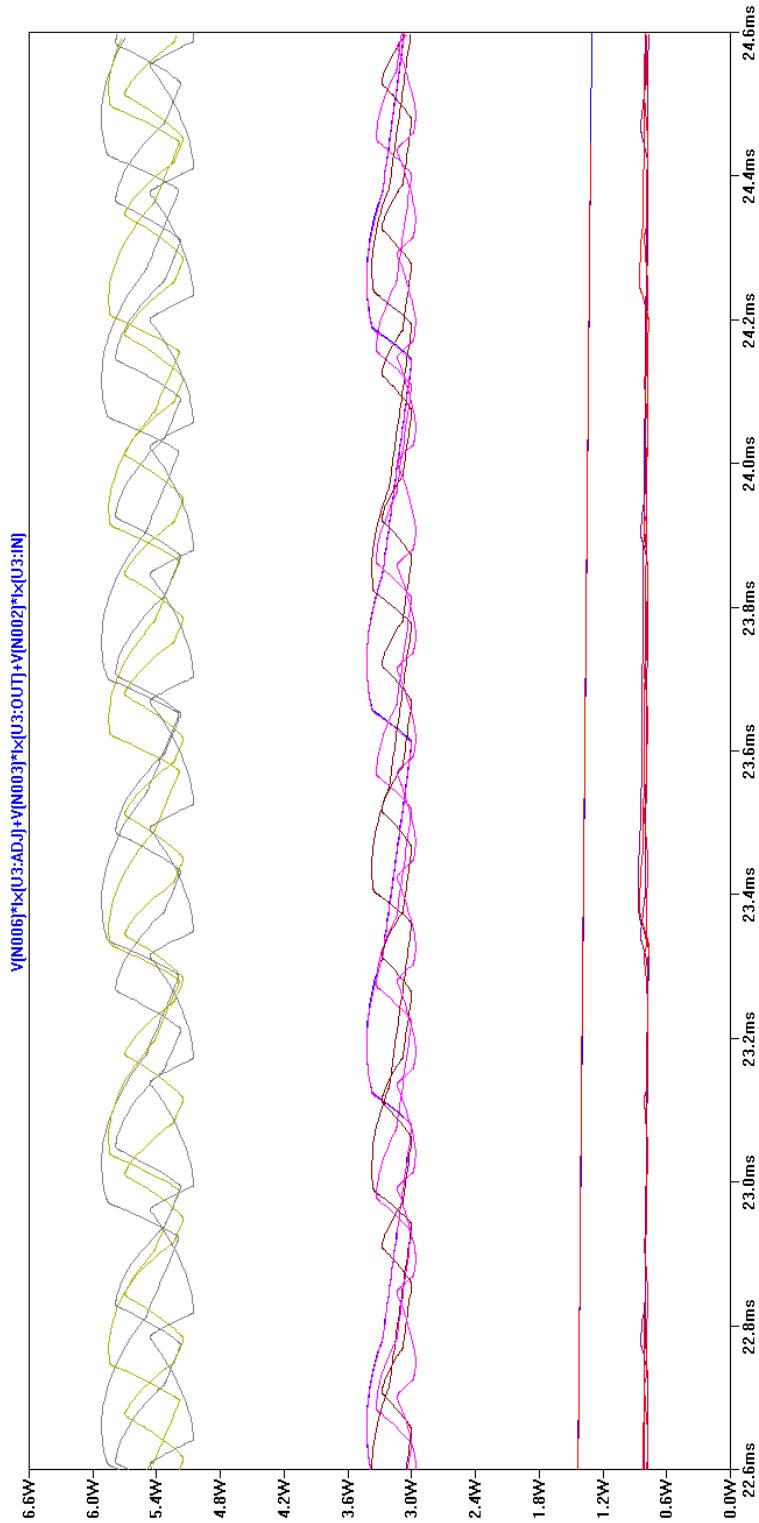


Figura 2.31: Potencia disipada por los reguladores.

2.3.4.2. Caracterización de la tarjeta.

Una vez simulado el comportamiento del circuito, se ha fabricado un primer prototipo de la tarjeta y comprobado su correcto funcionamiento. La tensión de entrada del regulador se ajusta a 1.7 V por encima de la tensión de salida del mismo para diferentes tensiones de salida del circuito rectificador. La tensión de salida se puede ajustar en un margen de 1.25 hasta 30 V mediante el potenciómetro multivuelta instalado. La regulación mediante el potenciómetro digital no se ha comprobado ya que no se ha terminado de montar la tarjeta controladora que gestiona el bus SPI y configura el potenciómetro.

En la figura 2.32 se representa en color marrón la tensión de rizado de salida medida y en color morado su transformada de Fourier cuando la tensión entregada por el circuito rectificador es de 15 V y el regulador entrega una corriente de 1.5 A a la resistencia de carga con una tensión de 5 V.

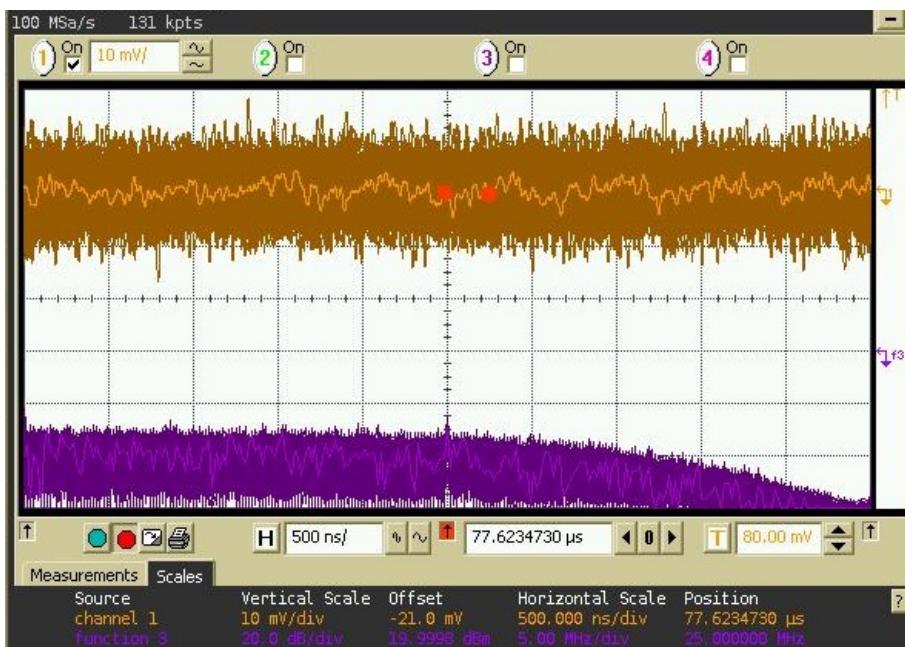


Figura 2.32: Tensión de rizado medida de la fuente de alimentación.

El valor de la tensión de rizado pico-pico para cualquier carga y tensión de salida (dentro de los límites de funcionamiento permitidos: $1.25 \leq V_{out} \leq 30$ V e $I \leq 7.5$ A) es inferior a 20 mV al igual que sucede en la gráfica 2.1 que se corresponde con los valores de la fuente de laboratorio tomados como objetivo del diseño.

2.3.5. Tarjeta controladora.

La tarjeta controladora, mostrada en la figura 2.33, es la encargada de monitorizar y controlar 6 tarjetas de regulación (slots 4-6 y 8-10). La última tarjeta de regulación, ubicada en el slot 11, no puede ser monitorizada por no disponer el conector DIN que une la tarjeta controladora con el *backplane* de pines suficientes. El control se realiza mediante un módulo de la marca Netburner y modelo MOD5213. Dicho módulo es similar al descrito en el apéndice B.1. El MOD5213 tiene un microcontrolador RISK de 32 bits a 66 MHz integrado que incorpora 8 conversores ADC (Analog to Digital Converter) de 12 bits.

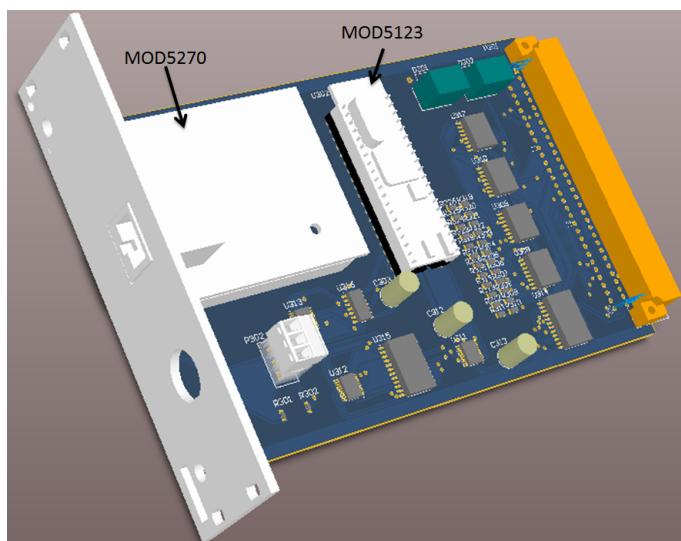


Figura 2.33: Tarjeta controladora.

Con estos conversores y la ayuda de un conjunto de multiplexores analógicos el microcontrolador es capaz de digitalizar la tensión y la corriente de salida de cada una de las 14 fuentes.

Además, controla los relés de salida de cada uno de los reguladores, permitiendo conectar o desconectar cada una de las tensiones generadas, y configura el valor de cada potenciómetro digital U302 mediante un bus I2C (véase la figura 2.21).

En el microcontrolador se ejecuta un software de gestión que monitoriza las fuentes y las hace trabajar conforme a la configuración programada (véase el apartado 2.3.6).

El módulo MOD5213 no incorpora conectividad IP. Por ello, la tarjeta controladora puede incorporar opcionalmente un módulo MOD5270 que facilita tal conectividad mediante un conector RJ45 que se refleja en la carátula frontal de la tarjeta controladora. Ambos módulos se comunican entre sí a través de un bus QSPI (Queued Serial Peripheral Interface).

Con este nuevo módulo la fuente podrá ser monitorizada y configurada desde la aplicación del sistema (véase el capítulo 5.2), no obstante, el software de control de la fuente en la aplicación del sistema aún no ha sido desarrollado.

La tarjeta controladora incorpora en su carátula frontal un pulsador que indica al microcontrolador si ha de desconectar o conectar las fuentes operativas. De esta forma el operador puede activar o desactivar localmente dichas fuentes. La tarjeta dispone también de 2 reguladores lineales para generar 12V y 3.3V. El regulador de 12 V genera una tensión que sirve para excitar las bobinas de los relés de cada regulador y para alimentar el amplificador acoplado a cada shunt de corriente. El segundo regulador de 3.3V proporciona la alimentación requerida por ambos microcontroladores. Los 2 reguladores se alimentan a través de la salida rectificada de la primera tarjeta rectificadora (véase la sección 2.3.3). La tarjeta ha sido fabricada y actualmente se encuentra en proceso de montaje.

2.3.6. Software de monitorización y control.

El software que se ejecuta en el MOD5213 configura el comportamiento de la fuente de tensión. Sus principales funciones se detallan en la tabla 2.11.

FUNCIONES
Ajustar la tensión de salida.
Activar el relé de salida tras la temporización programada.
Monitorizar la tensión y corriente.
Autoajustar la tensión en tiempo real.
Generar una alarma cuando se superan ciertos umbrales tras un determinado tiempo.
Tratar la alarma cuando se produzca (informar al PC, desconectar fuente etc...).
Determinar que fuentes se han de desconectar cuando una de ellas genera alarma.
Promediar las tensiones y corrientes almacenando valores máximos y mínimos.
Responder a las mensajes de configuración y de consulta de la aplicación del sistema.
Vigilar el pulsador del panel frontal para un encendido/apagado local.
Determinar el intervalo permitido de valores del potenciómetro digital.

Tabla 2.11: Funciones de la tarjeta controladora.

Los mayor parte de los parámetros son individuales para cada regulador. Un ejemplo de configuración de uno de los reguladores puede ser el siguiente:

- Ajustar la tensión hasta los 12 V.
- Activar el relé de salida tras 200 ms.
- Generar alarma #1 si durante 100 ms la tensión es superior a 12.3 V.

- Generar alarma #2 si durante 200 ms la tensión es inferior a 11.8 V.
- Generar alarma #3 si durante 100 ms la corriente es superior a 500 mA.
- Generar alarma #4 si durante 300 ms la corriente es inferior a 100 mA.
- Desactivar fuente y mandar mensaje UDP si se produce alarma #1.
- Mandar mensaje UDP si se produce alarma #2.
- Reducir tensión al 80 % si se produce alarma #3.
- Mandar mensaje UDP si se produce alarma #4.

Cuando una fuente se desconecta por causa de una alarma se puede determinar qué fuentes han de desconectarse también al mismo tiempo.

La aplicación del sistema deberá interrogar al microcontrolador de la tarjeta controladora cada vez que ésta quiera leer valores o parámetros del sistema de polarización. No obstante, cuando se produzca una situación de alarma se notificará a la aplicación del sistema la incidencia.

El software de control y monitorización de las fuentes ha sido escrito y depurado, pero quedan pendientes de realizar pruebas de interacción con el hardware. El software para la comunicación SPI con el microcontrolador MOD5270 y el código de este último para comunicarse con la aplicación del sistema tampoco han sido elaborados.

Capítulo 3

Sistema de antenas.

3.1. Descripción del sistema de antenas.

El sistema de antenas mostrado en la figura 3.2 ha sido diseñado en colaboración con la Universidad de Vigo en una configuración bifocal [21].

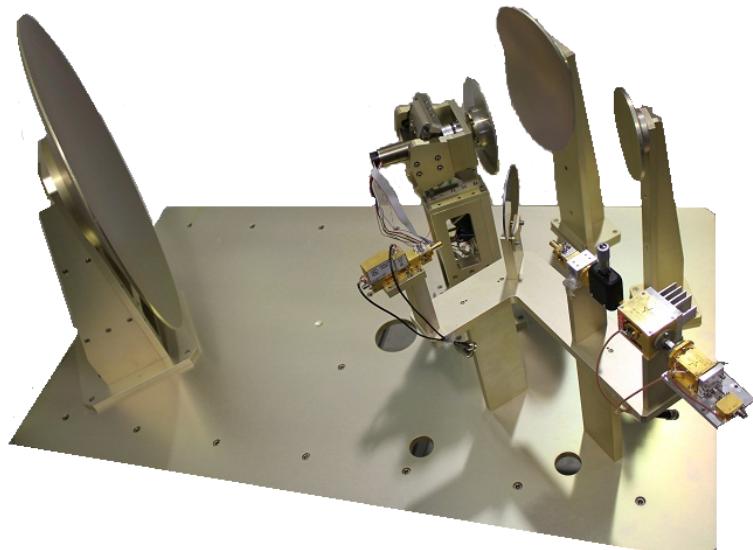


Figura 3.1: Sistema de antenas.

Este sistema de antenas ha sido valorado como el mejor diseño de antenas en el último congreso europeo EUCAP (European Conference on Antennas And Propagation) celebrado entre el 8 y el 12 de abril de 2013 en Göteborg (Suecia). Durante la fase de diseño se realizaron simulaciones previas del comportamiento de la antena mediante técnicas de óptica física. A continuación, se procede a realizar una breve descripción del mismo para una mejor comprensión del sistema.

El conjunto se compone de dos bocinas circulares (transmisora y receptora), un *Beam Splitter*, tres reflectores fijos y otro reflector orientable. Los reflectores y el tablero de la mesa que sustenta todo el equipamiento han sido mecanizados con aluminio aeronáutico de aleación 6082. En la figura 3.2 se enumeran cada uno de los elementos mencionados y en la tabla 3.1 se muestra la denominación de los mismos.

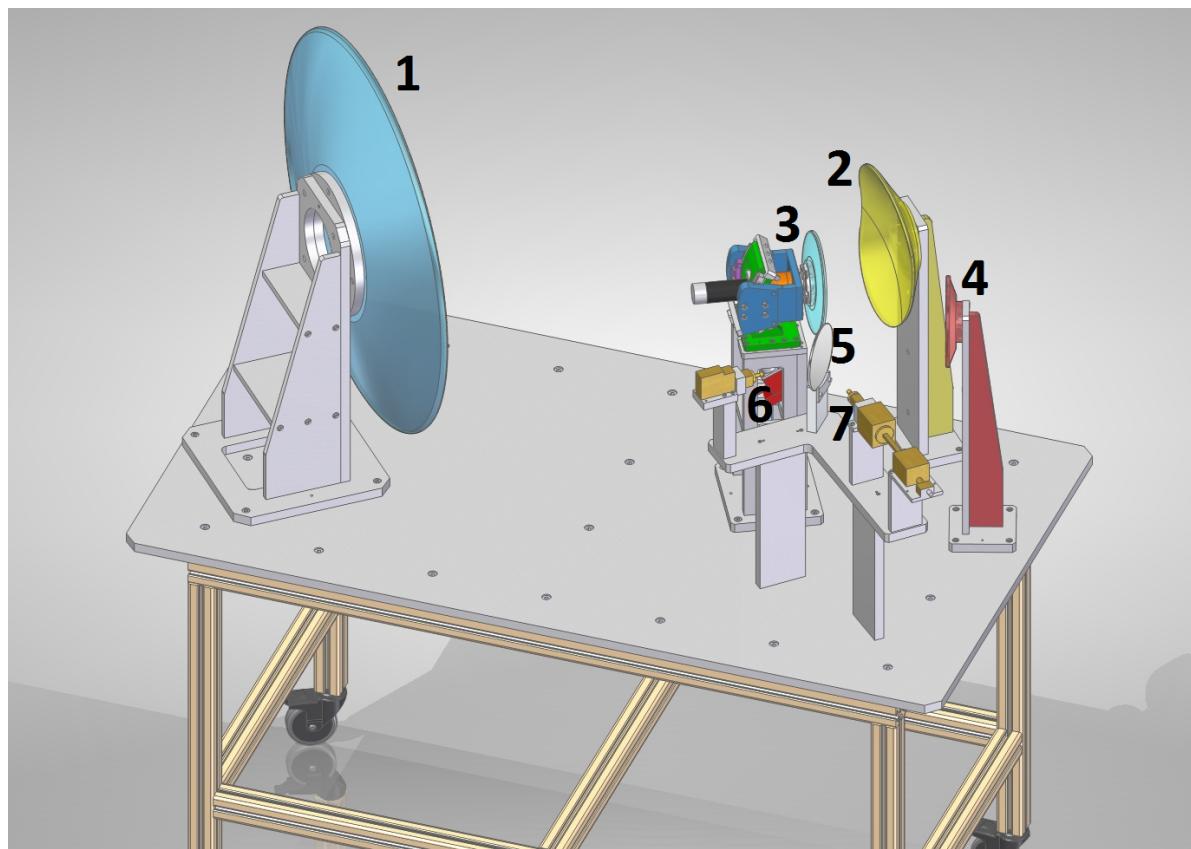


Figura 3.2: Sistema de antenas.

Elemento	Descripción
1	Reflector principal
2	Subreflector
3	Reflector orientable
4	Reflector de alimentación
5	Beam Splitter
6	Bocina transmisora
7	Bocina receptora

Tabla 3.1: Elementos del sistema de antenas.

Se describe a continuación una serie de conceptos preliminares como introducción. Posteriormente, estos mismos conceptos serán definidos formalmente en sus correspondientes apartados.

El sistema de antenas es capaz de explorar una volumen definido por un ortoedro cuyas dimensiones aproximadas se detallan en la figura 3.3. El ortoedro se encuentra a 8 metros de distancia del sistema de antenas. Su cara anterior se denomina **plano del objetivo**. El sistema de antenas es capaz, para una dirección de apuntamiento determinada, de iluminar en el *plano del objetivo* un área circular. Dicha área se denomina **spot** y su diámetro es aproximadamente de 1.3 cm.

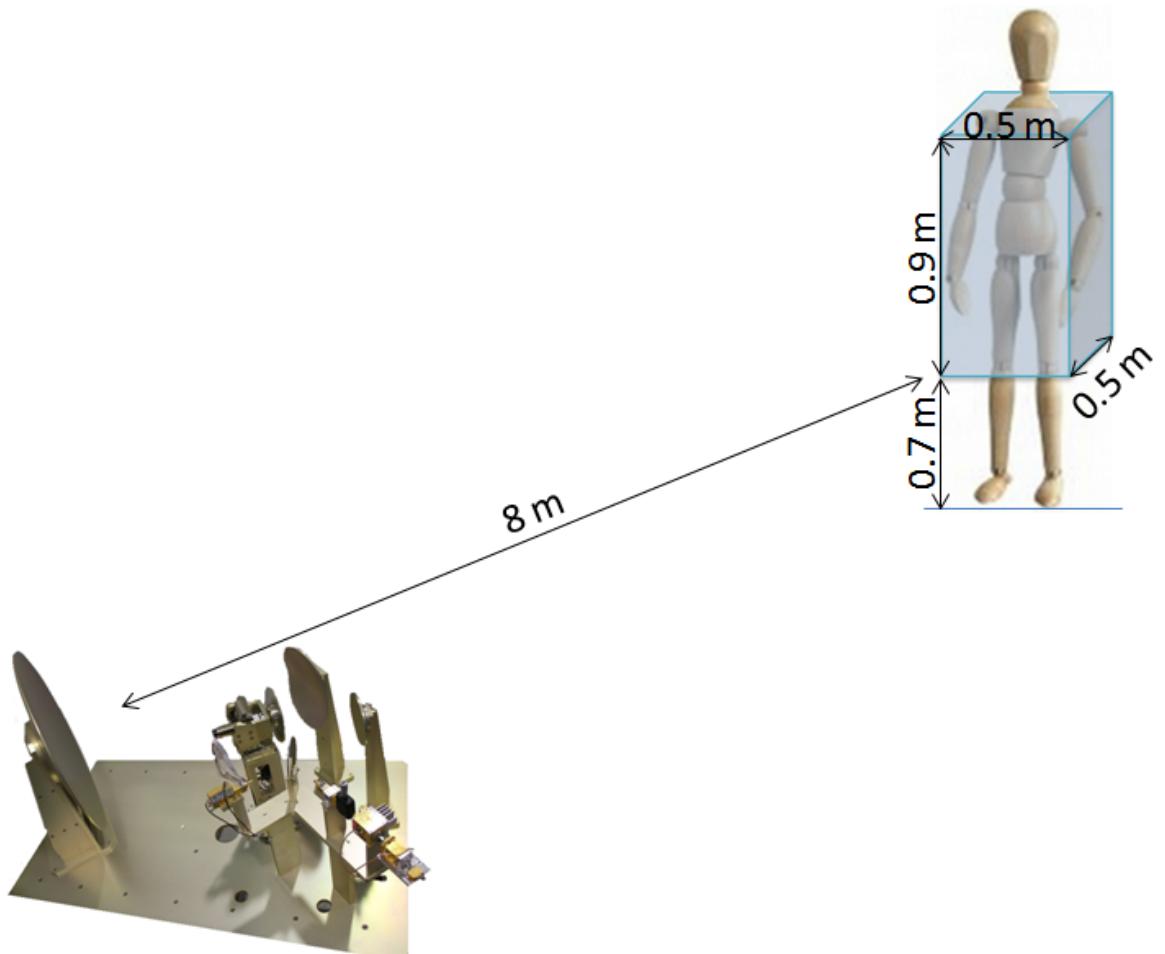


Figura 3.3: Volumen explorado por el sistema de antenas.

La dirección de apuntamiento del sistema de antenas se determina mediante el reflector orientable mostrado en la figura 3.4. Dicho reflector es denominado **espejo de exploración** por ser éste el elemento encargado de la exploración del *plano del objetivo*.

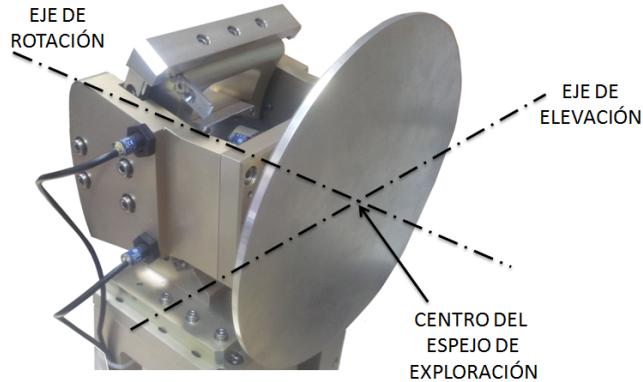


Figura 3.4: Espejo de exploración.

El *espejo de exploración* rota respecto de dos ejes, denominados **eje de rotación** y **eje de elevación**, para lograr la orientación adecuada. En la figura 3.5 se muestra un corte transversal de la estructura que sustenta el *espejo de exploración*. En color negro se representa el motor de rotación que hace girar al espejo (representado en azul claro) respecto del *eje de rotación*. El *eje de rotación* atraviesa el punto central del espejo y es capaz de adquirir una inclinación respecto del plano del suelo mediante la acción del motor de elevación (representado en color rojo). El ángulo que determina la inclinación del *eje de rotación* respecto del suelo se denomina ángulo γ . El *eje de elevación* es perpendicular al *eje de rotación*, también atraviesa el punto central del espejo y es paralelo al plano del suelo. El centro del *espejo de exploración* es un punto fijo en el espacio que no varía para cualquier rotación del espejo y cualquier valor γ .

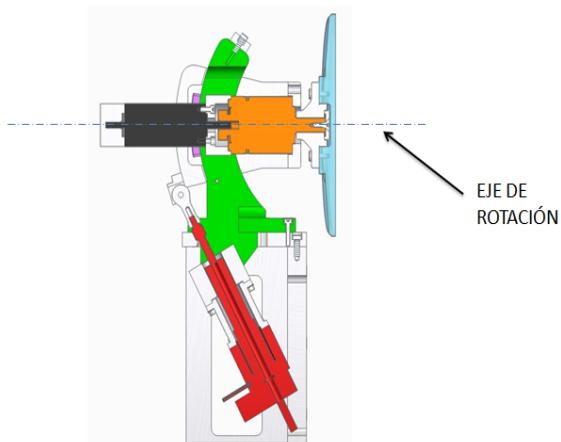


Figura 3.5: Eje de rotación con $\gamma = 0^\circ$.

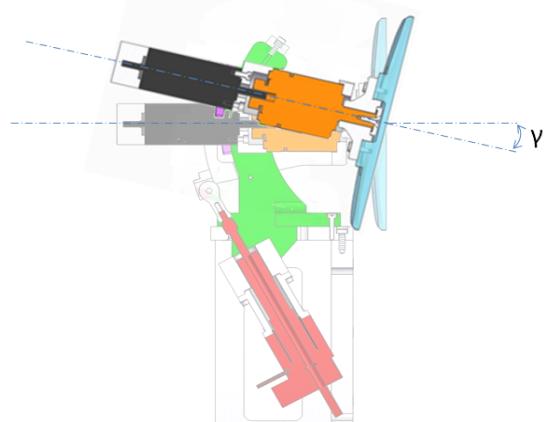


Figura 3.6: Eje de rotación con $\gamma = -6.8^\circ$.

El vector normal a la superficie del *espejo de exploración* forma un ángulo β con el *eje de rotación*. Ello es debido a la forma que presenta en su parte trasera y que se muestra en la figura 3.7. Por esta razón, cuando el espejo rota 360° respecto del *eje de rotación* el *spot* describe una elipse en el *plano del objetivo*. Según se muestra en la figura 3.8, realizando múltiples rotaciones con distintos valores de γ se logra barrer el *plano del objetivo*.

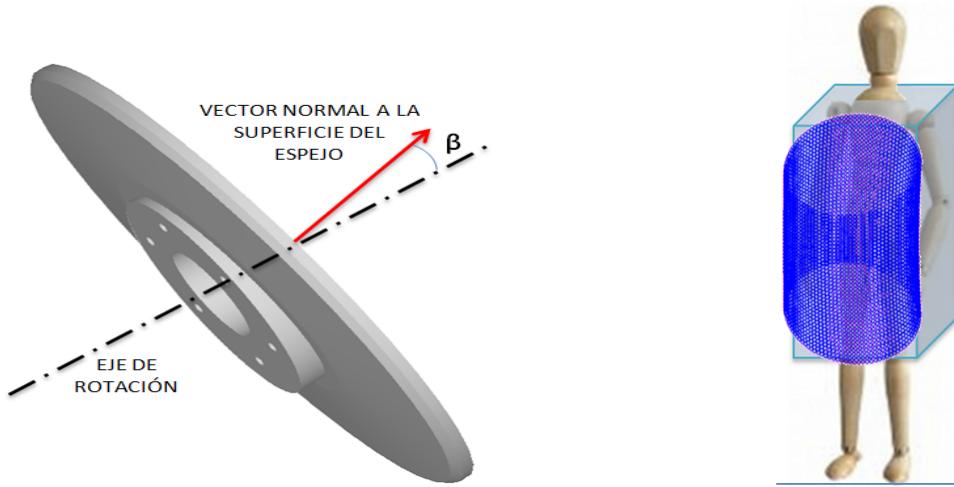


Figura 3.7: Detalle del espejo exploración.

Figura 3.8: Exploración del plano del objetivo.

La figura 3.9 define el sistema de coordenadas de la antena donde se han enumerado los elementos que aparecen representados según lo especificado en la tabla 3.1. El eje Z determina la dirección de propagación de las ondas.

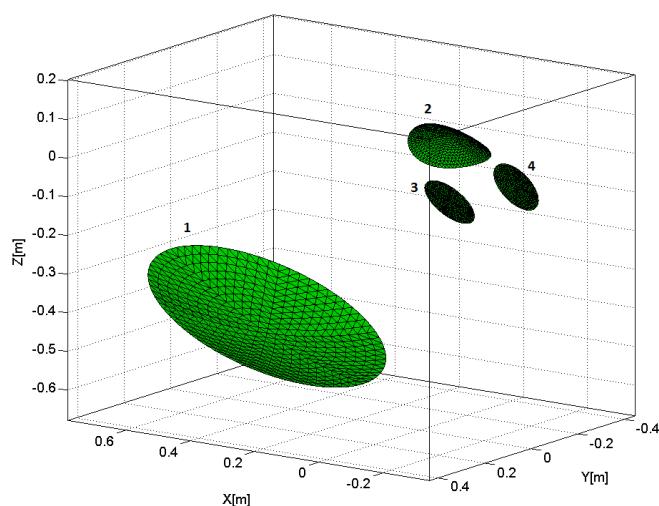


Figura 3.9: Sistema de coordenadas de la antena.

Para tratar de hacer más comprensible la descripción del funcionamiento de la antena, se tendrá en cuenta en una primera aproximación un diseño simplificado con las siguientes características:

- La superficie del subreflector se modela mediante un paraboloide elíptico cuyo foco es un punto clave en el diseño de la antena y se denomina **origen de antena**.
- La superficie del reflector principal describe un elipsoide (superficie cerrada de revolución cuya generatriz es una elipse). Su foco primario se ubica en el *origen de antena* (formando de esta manera una configuración confocal con el subreflector) y su foco secundario se localiza en la intersección entre el *plano del objetivo* y la directriz del elipsoide. Ambos focos distan entre si 8 metros.
- El *espejo de exploración* tiene un ángulo β de 0 grados, siendo β el ángulo definido entre el vector normal a la superficie del espejo y el *eje de rotación* (véase la figura 3.7).
- El ángulo de elevación γ tiene un valor de 0 grados, siendo γ el eje que permite reorientar la exploración de la antena verticalmente (véase la figura 3.6).

En la figura 3.10 se muestra una vista de planta del sistema de antenas y la disposición de los dos focos del elipsoide que modela el reflector principal.

El sistema radiante de este esquema simplificado ilumina como se describe a continuación en el sentido de la transmisión. La bocina transmisora (véase el apartado 3.1.7), cuyo centro de fase se ubica en las proximidades del foco del paraboloide que conforma el reflector de alimentación (véase el apartado 3.1.5), transmite con un diagrama de radiación de tipo pincel un frente de fase esférico. El 54 % de la potencia transmitida atraviesa el *beam splitter* (véase el apartado 3.1.6) e incide en el reflector de alimentación que proyecta, a modo de rayos paralelos (frente de fase plano), la potencia recibida hacia el *espejo de exploración*.

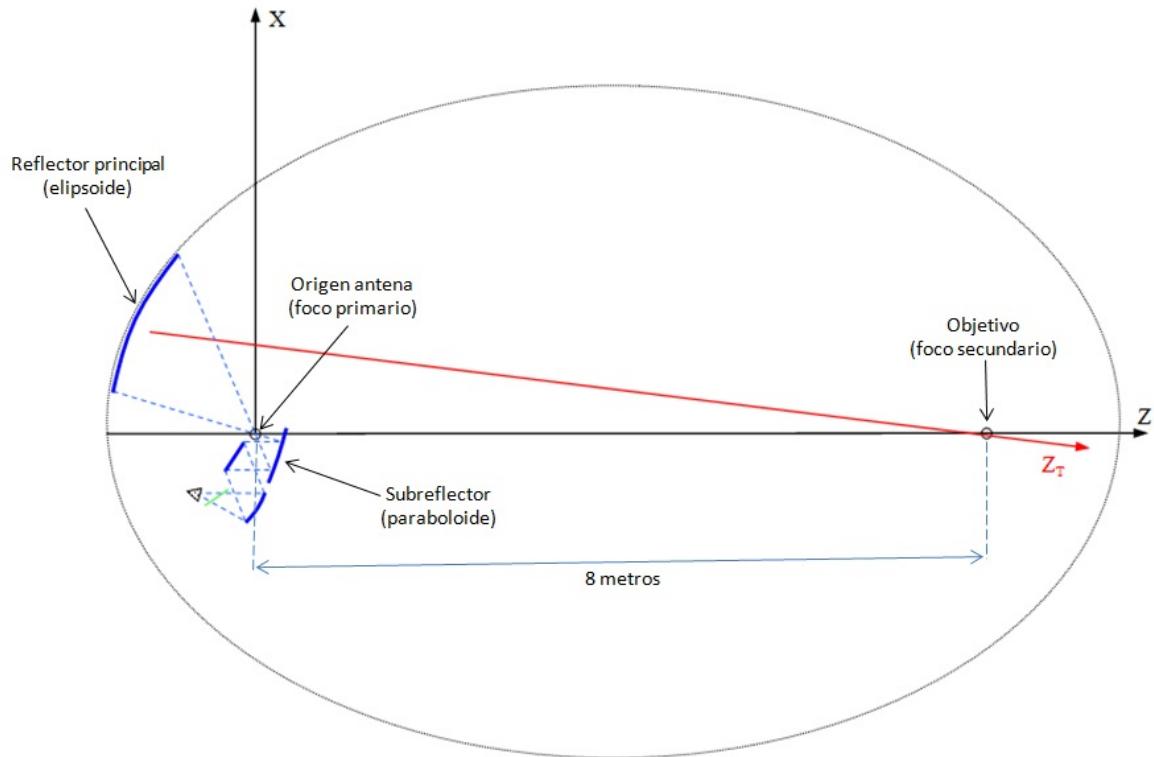


Figura 3.10: Elipsoide que modela el reflector principal.

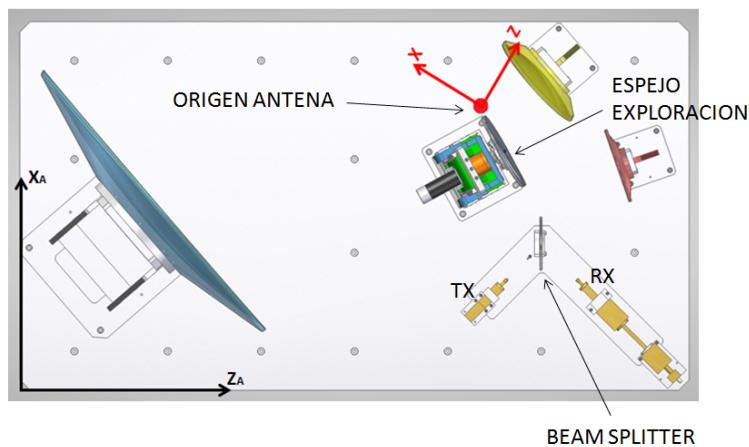


Figura 3.11: Vista de planta del sistema de antenas.

El *espejo de exploración* es una superficie plana y, por lo tanto, los rayos que emergen reflejados de éste en dirección al subreflector también son paralelos y con un frente de fase plano. Los rayos paralelos reflejados por el *espejo de exploración* inciden con una dirección similar a la de la directriz del paraboloide elíptico que modela el subreflector

y, por lo tanto, el frente de fase plano que incide en la superficie del reflector se refleja como un frente de fase esférico que converge en el *origen de antena*. El frente de fase esférico convergente se torna a divergente al atravesar el *origen de antena*.

El área iluminada en la superficie del reflector principal por el subreflector se determina mediante la intersección generada por la superficie del reflector con el volumen determinado por la superficie cónica que describe la cáustica de los rayos emergentes del subreflector. Puesto que los rayos que se reflejan en el reflector principal provienen del foco primario del elipsoide, los rayos reflejados convergen puntualmente en el foco secundario, donde está el objetivo de exploración del radar.

La configuración del reflector principal por parte del subreflector ha de ser de tipo *offset*, como se observa en la figura 3.10, para evitar bloqueos por parte del subreflector y de la estructura que sustenta el *espejo de exploración*.

La sección transversal de la cáustica de los rayos reflejados por el reflector principal va disminuyendo de tamaño hasta alcanzar el *plano del objetivo* (véase el apartado 3.1.8) el cual contiene el foco secundario, donde se producirá un máximo de iluminación ya previsible en la simulación de óptica física. El área iluminada en este plano, cuya densidad de potencia es superior a -3 dB respecto del valor máximo alcanzado, recibe el nombre de **spot** y su forma se aproxima a la de una circunferencia. En realidad, el *spot* no es un área sino un volumen de sección circular y más alargado en la dimensión Z.

Cuanto más alejados discurren los rayos de los focos primario y secundario del elipsoide, mayor es el área iluminada, la energía está más dispersa y más ancho es el *spot*, esto es, de menor calidad. Esto sucede cuando la antena explora y se separa de la situación ideal descrita hasta aquí.

En el modelo real del sistema de antenas, el *espejo de exploración* presenta una inclinación respecto del eje de rotación ($\beta \neq 0$, véase la figura 3.7) que modifica la dirección de reflexión de los rayos. Gracias a esta nueva orientación, los rayos que emergen del espejo no son paralelos a la directriz del paraboloide y, por este motivo, no se concentran entorno al *origen de antena*.

En esta nueva situación, los rayos se cruzan en un punto próximo al *origen de antena* denominado *punto de cruce*, y cada rayo llega a dicho punto de convergencia con una fase distinta. Cuando el espejo rota 360° respecto del *eje de rotación* (véase 3.1.4), el *punto de cruce* se desplaza describiendo una trayectoria con forma aproximada de elipse alrededor del *origen de antena*. La razón por la cual la trayectoria descrita no es circular se debe a la configuración *offset* de los reflectores impuesta para evitar bloqueos.

Un fenómeno similar a lo que ocurre en torno al *origen de antena* sucede alrededor del foco secundario del elipsoide. La rotación completa del espejo provoca que el *spot* describa una trayectoria casi elíptica en el *plano del objetivo* (véase 3.1.8).

Dicha elipse tiene un eje mayor, casi paralelo al plano del suelo de aproximadamente 50 cm y un eje menor, casi perpendicular al mismo plano de unos 40 cm. El área y la forma del *spot* sufren variaciones según el ángulo de rotación ϕ del espejo. Si además de

rotar 360° se modifica el ángulo de elevación γ (véase 3.2) la trayectoria descrita sufre una traslación vertical en el mismo *plano del objetivo* y una ligera deformación respecto de la forma descrita en $\gamma = 0^\circ$; de forma que el paralelismo y la perpendicularidad de sus ejes mayor y menor respectivamente se pierden. Además, por el mismo motivo, el *spot* modifica aún más su forma pareciéndose más a una elipse.

La figura 3.12 representa (según el sistema de coordenadas determinado en la figura 3.9) la trayectoria descrita por los *spots* según el ángulo de inclinación del espejo. Dicho ángulo β viene determinado por la forma del conjunto del espejo y, por lo tanto, no es variable. Aun así, se representan las curvas para distintos valores de éste para una mejor comprensión.

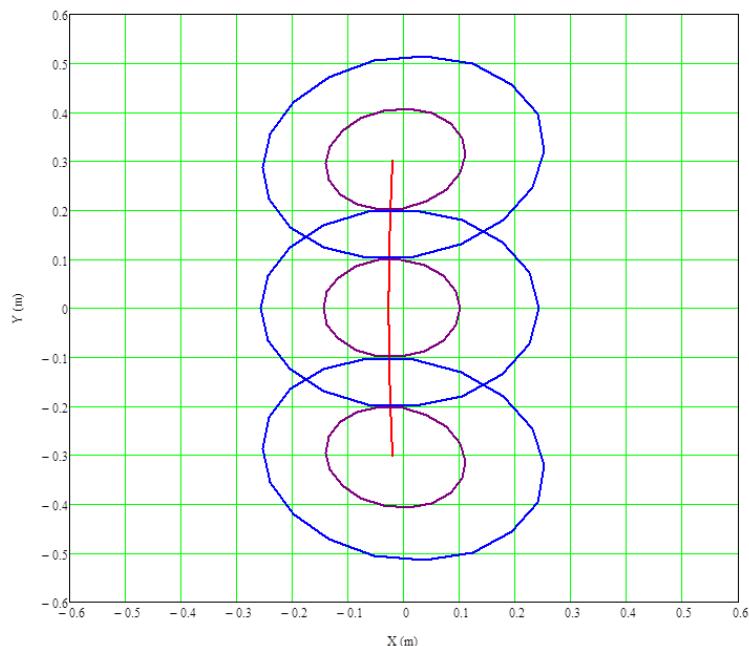


Figura 3.12: Trayectoria descrita por el *spot* en el plano del objetivo.

En la gráfica se representan en color azul las curvas descritas para el valor de β real (4.5°) en tres posiciones de apuntamiento en elevación distintas: $\gamma = +6.8^\circ$, $\gamma = 0^\circ$ y $\gamma = -6.8^\circ$ que se corresponden respectivamente con la elipse cuyos puntos tienen unas componentes Y negativas, la elipse cuyos puntos tiene componentes Y de distinto signo y la elipse cuyos puntos tienen unas componentes Y positivas. De forma análoga, se representan las mismas curvas en color morado para un valor de $\beta = \frac{\beta}{2}$.

Por último, en color rojo se representan los puntos (y no curvas cerradas a pesar de la variación en ϕ de 360 grados conforme a lo explicado anteriormente en una primera aproximación) para un valor de β nulo y los valores de γ comprendidos en el intervalo [-6.8, 6.8] grados.

La configuración *offset* es la razón por la cual la curva representada en color rojo no es una recta vertical. Debido al mismo principio, el movimiento circular descrito por

el espejo se transforma en una trayectoria casi elíptica. Incluso para un valor de $\gamma = 0$ grados la elipse está ligeramente deformada.

3.1.1. Reflector principal.

El reflector principal (figura 3.13) recibe la potencia transmitida por el subreflector (véase 3.1.2) y la proyecta sobre el *plano del objetivo* (véase 3.1.8).

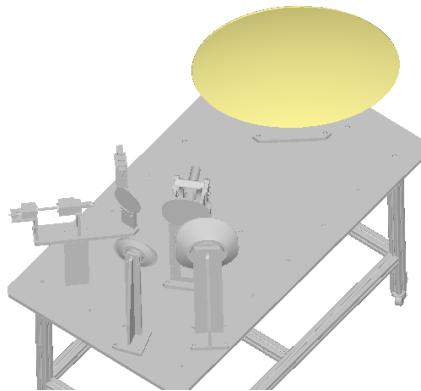


Figura 3.13: Reflector principal.

La superficie del reflector es parte de un elipsoide (superficie cerrada de revolución cuya generatriz es una elipse) descrito mediante la ecuación 3.1. Uno de los focos del elipsoide, el más próximo al reflector, coincide con el foco del subreflector conformando ambos un conjunto *confocal*. Este foco se denomina *foco primario*. El foco restante se ubica a 8 metros del *foco primario* y se denomina *foco secundario*. El foco secundario pertenece al *plano del objetivo*.

$$Z_{m1}(x, y) = C - A \sqrt{1 - \frac{x^2 + y^2}{B^2}} \quad (3.1)$$

En el diseño final esta superficie es distorsionada por un polinomio de cuarto orden según se muestra en la ecuación 3.2 para lograr una configuración bifocal [21] (véase el apartado 3.1.3).

$$Z_{m2}(x, y) = a_0 + a_1(x^2 + y^2) + a_2(x^2 + y^2)^2 \quad (3.2)$$

La superficie queda finalmente definida según se indica en 3.3 la cual es muy similar a la descrita por el elipsoide original descrito en la ecuación 3.1.

$$Z_m(x, y) = Z_{m1}(x, y) + Z_{m2}(x, y) \quad (3.3)$$

En la tabla 3.2 se muestran los valores de las constantes que describe la superficie resultante.

Parámetro	Descripción	Valor
x	Coordenada x de la superficie	m
y	Coordenada y de la superficie	m
A	Semieje mayor de la elipse generatriz	4.6 m
B	Semieje menor de la elipse generatriz	2.271563 m
C	50 % de la distancia interfocal	4 m
a_0	Coeficiente de distorsión constante	$2.768 \cdot 10^{-6} \approx 0$ m
a_1	Coeficiente de distorsión cuadrática	$1.516875 \cdot 10^{-3}$ m ⁻¹
a_2	Coeficiente de distorsión de cuarto orden	$9.715516 \cdot 10^{-4}$ m ⁻³

Tabla 3.2: Descripción de la superficie del reflector principal.

3.1.2. Subreflector.

El subreflector (figura 3.14) recibe la potencia radiada por el espejo de exploración y la proyecta sobre el reflector principal haciéndola pasar por el *origen de antena*.

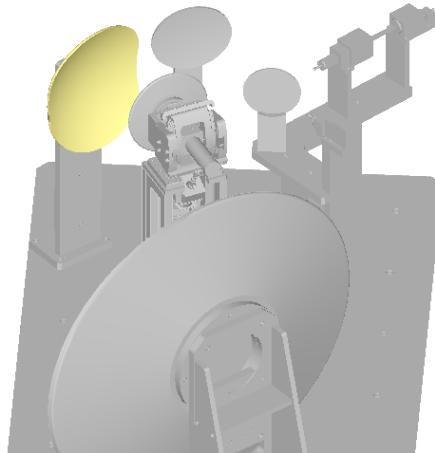


Figura 3.14: Subreflector.

La superficie del subreflector se modela mediante un parabolóide (ecuación 3.4) cuyo foco coincide con el foco primario del reflector principal haciendo de esta manera que el conjunto tenga una configuración *confocal*.

$$Z_{s1}(x, y) = b_0 + b_1(x^2 + y^2) \quad (3.4)$$

Al igual que el reflector principal, la superficie está modificada por un polinomio de cuarto orden (ecuación 3.5) que modifica la posición del foco del parabolóide logrando una configuración bifocal (véase 3.1.3).

$$Z_{s2}(x, y) = b_2(x^2 + y^2)^2 \quad (3.5)$$

La superficie del subreflector resultante se describe según la ecuación 3.6.

$$Z_s(x, y) = Z_{s1}(x, y) + Z_{s2}(x, y) \quad (3.6)$$

En la tabla 3.3 se muestran los valores de cada uno de los parámetros que modelan el subreflector.

Parámetro	Descripción	Valor
x	Coordenada x de la superficie	m
y	Coordenada y de la superficie	m
b_0	Coordenada Z del vértice	0.109091 m
b_1	Coeficiente del paraboloide	-2.291119 m^{-1}
b_2	Coeficiente de distorsión de cuarto orden	-0.420786 m^{-3}

Tabla 3.3: Descripción de la superficie del subreflector.

3.1.3. Configuración bifocal.

Según lo explicado en 3.1, cuanto más alejados discurren los rayos de los focos primario y secundario del elipsoide, mayor es el área iluminada en el *plano del objetivo*. Esto provoca a su vez que estos mismos rayos presenten un mayor desfase entre ellos.

Cuando los rayos alcanzan el *plano del objetivo* con una fase similar (PLRMS bajo. Véase la figura 3.17) se produce en el interior del spot una contribución de amplitudes de cada rayo que maximiza la densidad de potencia de forma crítica en torno al centro del spot. Como resultado de este fenómeno, en una zona del *spot* concreta se produce un pico alto de densidad de potencia que reduce el área del *spot* definida a -3 dB. De tal forma, el tamaño del *spot* depende de las coordenadas de apuntamiento del sistema de antenas en el *plano del objetivo*.

Para atenuar este fenómeno, se ha modificado la forma del reflector principal y del subreflector. Ambas superficies han sido conformadas por sendos polinomios de cuarto orden (véase los apartados 3.1.2 y 3.1.1). Con ello se logra una configuración bifocal que desplaza el foco secundario original del elipsoide.

Por tratarse de superficies de revolución, el desplazamiento del foco consigue que las mismas propiedades de enfoque que se tenían en el foco secundario ahora se produzcan a lo largo de una circunferencia de 30 cm de radio alrededor del foco original. Dicha circunferencia se denomina **anillo focal**.

El *anillo focal* está contenido en un plano paralelo al plano $X_T Y_T$ de la antena denominado **plano focal** que está ligeramente inclinado respecto del *plano del objetivo* debido a la configuración *offset*, figura 3.15. Se corta con éste en dos puntos situados sobre un eje paralelo al eje Y de la antena para un valor Z de 8 metros. Esto es, los dos puntos del *plano de objetivo* situados verticalmente a ± 30 cm del foco secundario original.

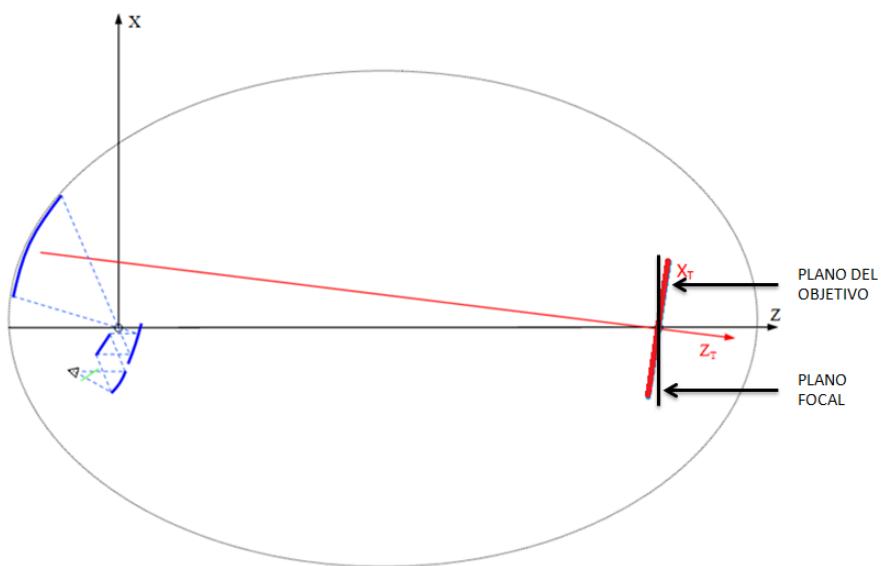


Figura 3.15: Orientación del plano focal.

Cuando el sistema de antenas apunta a cualquier punto del anillo focal, todos los rayos llegan a éste con un error de fase nulo. Fuera del anillo focal, los puntos de error de fase mínimo (distínganse de los *spots* que se definen únicamente en el *plano del objetivo*) para cada posición de exploración se localizan en una curva de forma aproximadamente parabólica cuyo vértice, ubicado entre la antena y el *plano del objetivo*, se encuentra a unos 30 cm del *plano del objetivo*. El foco parabólico de cada una de dichas curvas se sitúa entre la antena y su vértice.

En la figura 3.16 se muestra en color azul un rectángulo que representa la escena que se va a explorar. En color negro se representa la circunferencia del *anillo focal* y, por último, se muestran los *spots* del barrido de la escena. Éstos últimos se representan con un color que es dependiente de la distancia existente expresada en metros entre cada *spot* y el foco secundario, según se muestra en la barra de color adjunta.

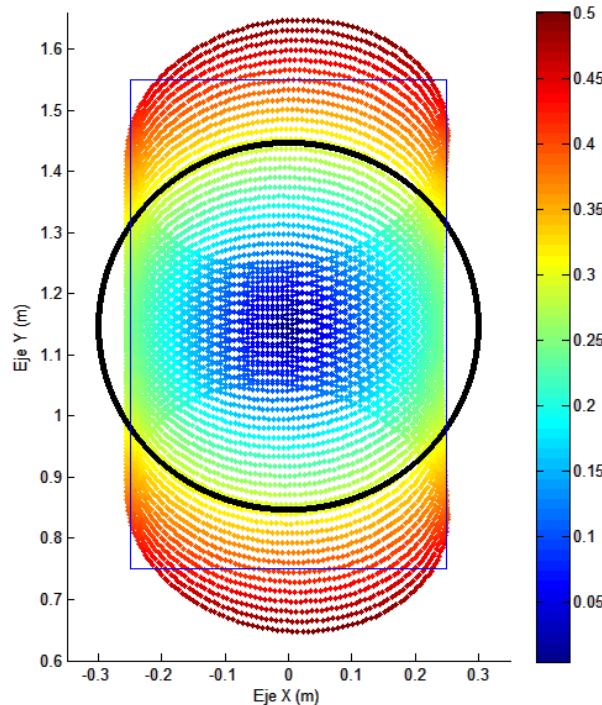


Figura 3.16: Distancia de los *spots* al anillo focal.

La figura 3.17 muestra el desfase medio existente entre todos los rayos del área iluminada en el *plano del objetivo*. Cada curva cerrada representa una rotación completa en ϕ para diferentes valores de γ . Esta gráfica muestra cómo los *spots* que quedan dentro del *anillo focal* tienen un desfase medio menor y, por lo tanto, su tamaño es menor. Los *spots* dibujados en color amarillo (distancia = 0.3 metros) en la figura 3.16 son los que tienen un menor tamaño. Los *spots* que tienen un color naranja o rojo son los *spots* de mayor tamaño.

En conjunto, al discurrir los *spots* en las inmediaciones del anillo focal, se consigue una mejora en el "enfoque medio" sobre el *plano focal* respecto de la configuración *confocal* original, en la que el anillo focal es simplemente un punto. En este último caso, la distancia media de separación de los *spots* a este único punto es mayor.

Se ha realizado una simulación de óptica física para tratar de estimar el tamaño del *spot*. La simulación analiza 8 puntos de la elipse que se genera mediante un giro completo del espejo de exploración con un valor γ de 0 grados. La figura 3.18 muestra la variación del campo eléctrico respecto del valor máximo detectado en los 8 puntos expresada en dB's.

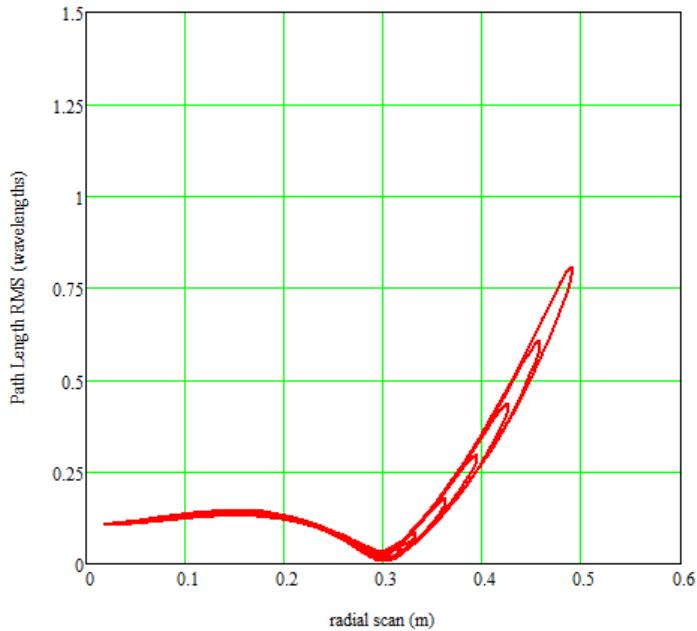


Figura 3.17: Desfase medio RMS según distancia al foco secundario.

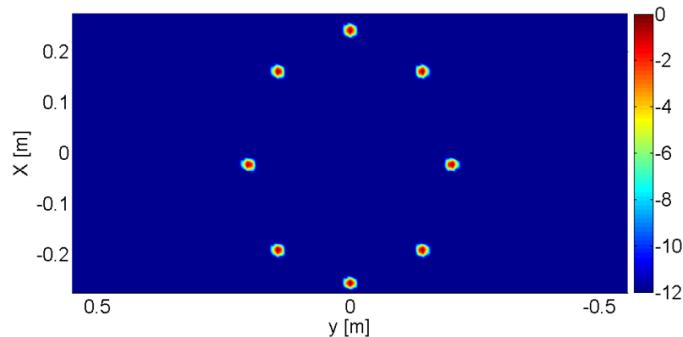


Figura 3.18: Valor del campo relativo en 8 puntos de la elipse central.

La figura 3.19 representa en color azul la diferencia expresada en dB's entre el máximo obtenido en los 8 puntos analizados y el máximo de cada uno de los puntos analizados. En color rojo se expresa el tamaño de cada uno de los 8 *spots*. El tamaño se determina siguiendo el criterio de corte a -3 dB respecto del valor máximo alcanzado en cada *spot*.

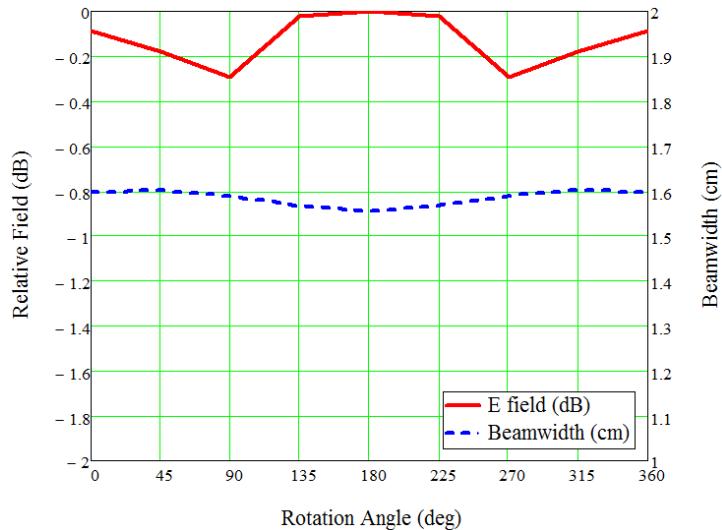


Figura 3.19: Tamaño del *spot* en 8 puntos de la elipse central.

De manera análoga se ha realizado la misma simulación en la elipse generada con un valor gamma de -6.8 grados cuyos resultados se muestran en las gráficas 3.20 y 3.21.

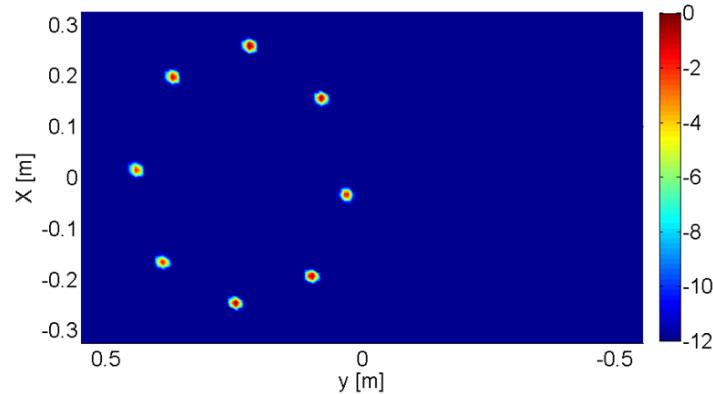


Figura 3.20: Valor del campo relativo en 8 puntos de la elipse superior.

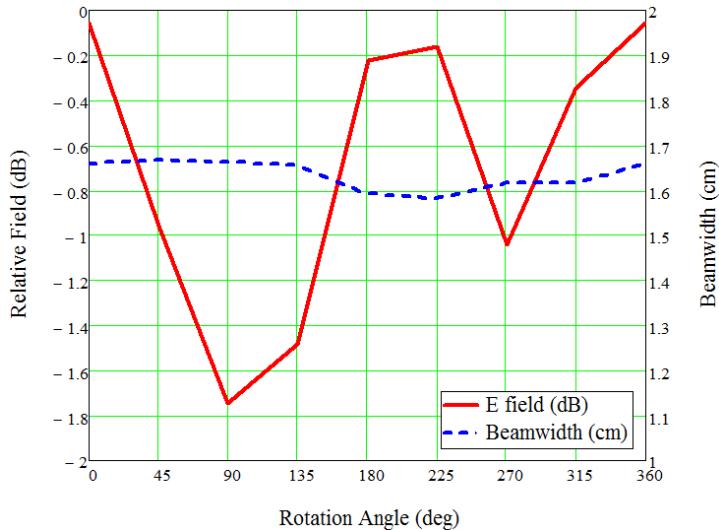


Figura 3.21: Tamaño del *spot* en 8 puntos de la elipse superior.

3.1.4. Espejo de exploración.

El conjunto del espejo (figura 3.22) se compone de dos figuras geométricas unidas entre sí y que conforman una única pieza de aluminio: El espejo de rotación, el cual es un objeto de revolución, y el segmento cilíndrico que se integra en su parte trasera.

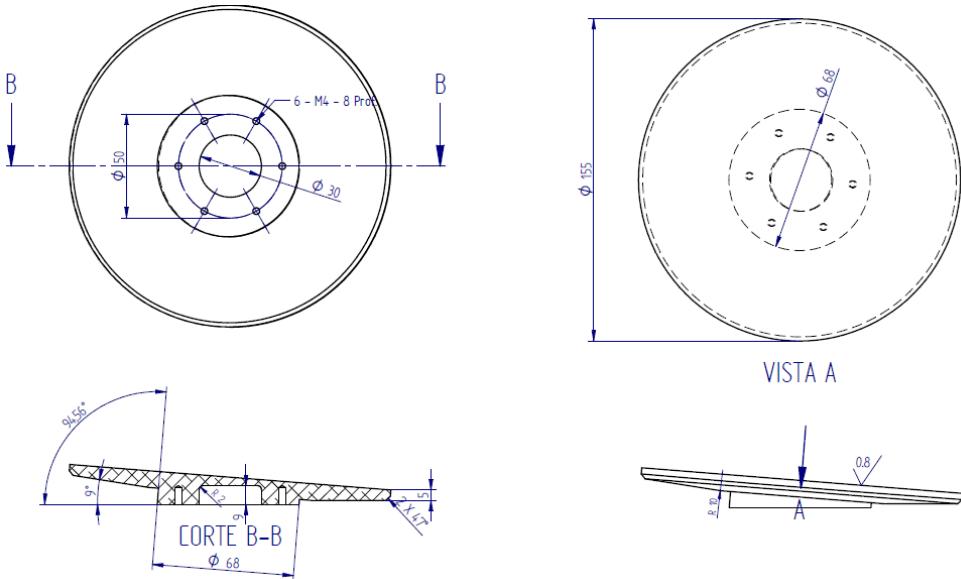


Figura 3.22: Espejo de exploración.

Una de las bases del segmento cilíndrico es circular y se acopla a la reductora del motor de rotación (véase el apartado 3.2.2.1). La otra base del segmento está definida

por un plano oblicuo respecto de la base circular y forma un ángulo β de 4.557198 grados con ésta. El **ángulo de rotación** ϕ del *espejo de exploración* se define como el ángulo recorrido por el motor de rotación a partir de una posición determinada del mismo denominada $\phi = 0$.

$\phi = 0$ es aquella posición del espejo en la cual la normal al mismo refleja la potencia recibida por el *feed reflector* en la coordenada X más negativa del *subreflector*. En esta posición, y cuando γ tiene el valor de 0 grados, el vector normal a la superficie del espejo es además paralelo al plano que contiene el tablero de la mesa que sustenta todo el sistema y al suelo. En $\phi = 0$ el sistema de antenas ubica el *spot* en aquellos puntos del plano del objetivo (véase el apartado 3.1.8) cuya coordenada X es más negativa.

La estructura que soporta el motor de rotación (figura 3.23) impone unos límites en la variación de γ de ± 10.1 grados. No obstante, el sistema de antenas es capaz de enfocar correctamente hasta una variación máxima de γ de ± 6.8 grados, por lo que los límites de la estructura no limitan la capacidad de apuntamiento del sistema.

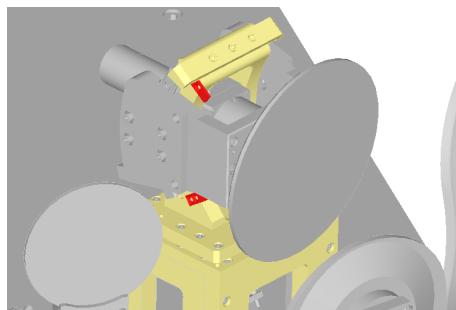


Figura 3.23: Estructura soporte del espejo de exploración.

Cuando γ toma el valor de $+6.8$ grados, el sistema de antenas explora la parte superior del *plano del objetivo*. El centro de masas del conjunto de piezas, que conforman el espejo de 526 gramos de peso, está desplazado 0.54 mm respecto del eje de rotación. Este hecho produce vibraciones en la estructura cuando la velocidad de giro es superior a 2,500 rpm.

3.1.5. Reflector de alimentación.

El reflector de alimentación (figura 3.24) refleja el frente de fase esférico transmitido por la bocina transmisora como un frente de fase plano hacia el *espejo de exploración*.

La superficie del reflector de alimentación se modela mediante un paraboloide elíptico según la ecuación 3.7.

$$Z_f(x, y) = c_0(x^2 + y^2) \quad (3.7)$$

En la tabla 3.4 se muestran los parámetros que modelan la superficie.

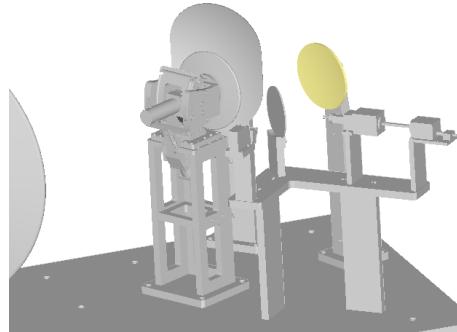


Figura 3.24: Reflector de alimentación.

Parámetro	Descripción	Valor
x	Coordenada x de la superficie	m
y	Coordenada y de la superficie	m
c_0	Coeficiente del paraboloide	0.25 m

Tabla 3.4: Descripción de la superficie del reflector de alimentación.

3.1.6. Beam splitter.

El *Beam Splitter* (figura 3.25) es un segmento cilíndrico del fabricante Reflex USA de 10.16 cm de diámetro y 3 mm de grosor compuesto de silicio, cuya función es la de desacoplar el transmisor del receptor. Su índice de refracción a 300 GHz es de 3.416.

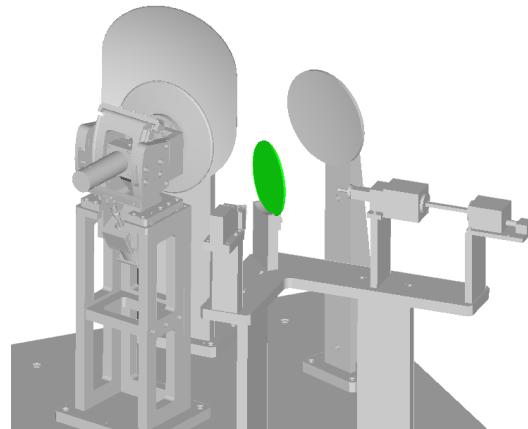


Figura 3.25: *Beamsplitter*.

La base circular del segmento cilíndrico forma un ángulo de 45 grados con la dirección de propagación de las ondas transmitidas. El 56 % de la potencia incidente atraviesa el *Beam Splitter* y el 44 % restante se refleja con un ángulo de 90 grados respecto de la dirección de propagación de las ondas. Lo mismo sucede en la otra cara con la incidencia de las ondas reflejadas según se muestra en la figura 3.26.

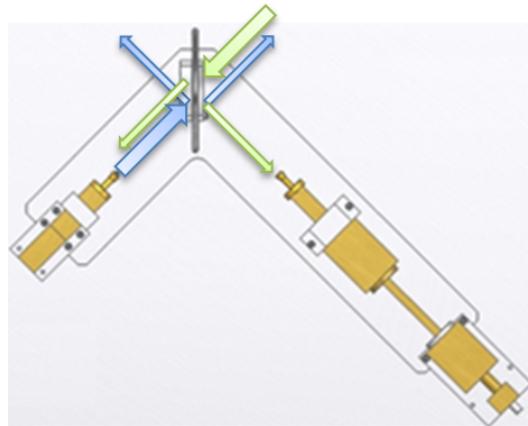


Figura 3.26: Descripción del *Beamsplitter*.

3.1.7. Bocinas.

Las bocinas (figura 3.27) transmisora (amarillo) y receptora (verde) son de la marca Radiometer Physics GMGH y modelo FH-PP-300. Son bocinas *dual-mode* de tipo Potter.

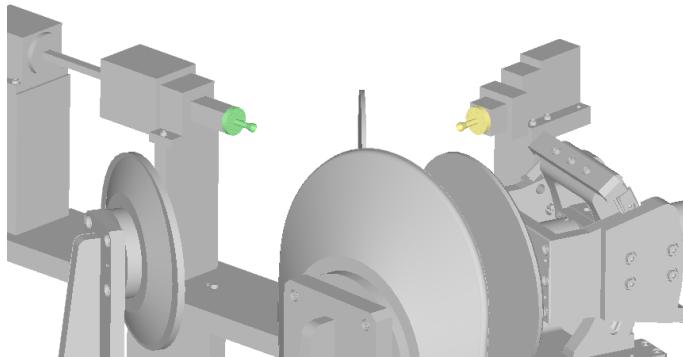


Figura 3.27: Bocinas.

Las bocinas *dual-mode*, al igual que las bocinas corrugadas, logran transmitir un modo con polarización vertical similar al HE_{11} . Son bocinas más sencillas de mecanizar que las bocinas corrugadas aunque su ancho de banda es menor.

Las bocinas de este tipo excitan desde el final de la guía una mezcla de modos $TE_{11} + TM_{11}$ que generan en conjunto, tras una compensación final en la sección final de la bocina, un modo similar al HE_{11} según se muestra en la figura 3.28.

Este modelo de bocina tiene su centro de fase en el interior de la misma a 6 mm respecto de la apertura, la cual es de 6.4 mm de diámetro. La bocina tiene un ancho de haz a -10 dB de 23.5 grados.

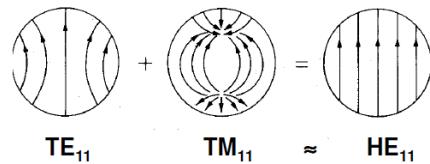


Figura 3.28: Excitación de modos de la bocina tipo *Potter*.

3.1.8. Plano del objetivo.

El objetivo del sistema de antenas es el de explorar una escena rectangular cuyas dimensiones son 50 cm de ancho por 90 cm de alto. La escena está centrada a una altura de 1.147 metros desde el suelo y dista aproximadamente 8 metros desde el reflector principal. La orientación del **plano del objetivo** no es totalmente perpendicular al eje Z como se muestra en la figura 3.29. Esto es debido a la configuración en *offset* del sistema de antenas necesaria para evitar bloqueos entre los distintos reflectores.

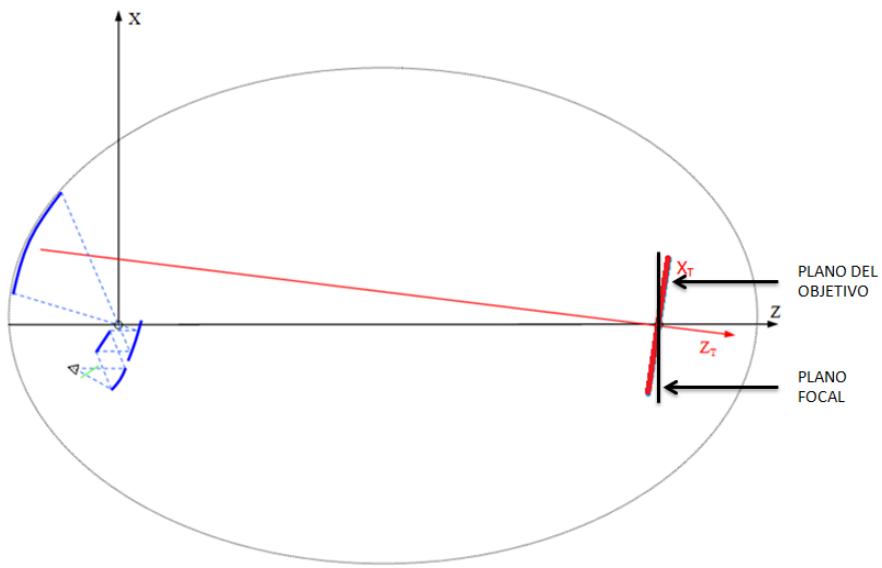


Figura 3.29: Orientación del eje X_T del plano del objetivo.

El *plano del objetivo* presenta una rotación τ respecto del eje Z de 3.278 grados, es perpendicular al eje Z_T definido en la figura 3.10 y contiene al foco secundario del elipsoide que conforma el reflector principal (véase 3.1.1). El sistema de coordenadas del *plano del objetivo* es bidimensional. Lo componen el eje Y_T , que es perpendicular al suelo, y el eje X_T , que es paralelo a éste. El origen está definido en una recta perpendicular al suelo que contiene el foco secundario y dista del suelo 1.147 metros.

Desde una vista obtenida desde el reflector principal, el valor de la componente X_T disminuye hacia la izquierda y se incrementa hacia la derecha. El vector unitario Y_T viene determinado por el producto vectorial $\hat{Z}_T \times \hat{X}_T$

3.1.9. Movimiento descrito por el sistema de antenas.

Para lograr un barrido completo de la escena que se pretende capturar, se debe realizar un movimiento compuesto que se logra mediante la actuación de los motores de rotación y elevación (véase la sección 3.2). El control de dicho barrido forma parte del alcance del proyecto ya que interactúa con el sistema de control y sincronismo del sistema. En este apartado no se determinan cuales son las velocidades óptimas ni máximas de cada uno de los motores para una correcta exploración del plano del objetivo. Tan sólo se pretende describir la trayectoria descrita por la dirección de apuntamiento cuando ambos motores trabajan simultáneamente.

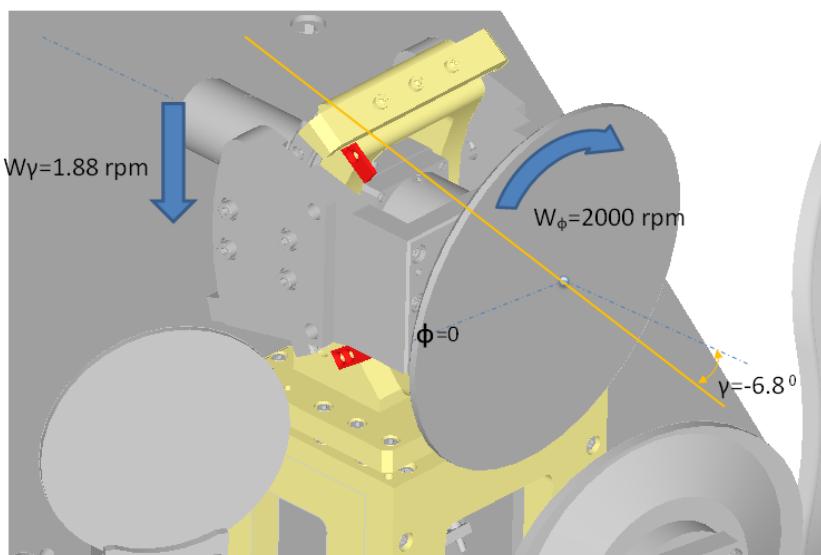


Figura 3.30: Movimiento del espejo de exploración.

Para comenzar un barrido de la escena, se ha de posicionar el espejo de tal forma que el ángulo de elevación γ tome el valor de -6.8 grados. El motor de rotación comienza a girar hasta alcanzar una velocidad de giro constante. Durante este periodo, el *spot* describe la elipse más alta que el sistema es capaz de apuntar.

Una vez alcanzada la velocidad deseada se programa el circuito detector de la posición angular $\phi = 0$ (véase el apartado 5.1.3). Cuando el espejo alcanza dicha posición $\phi = 0$, el circuito detector de inicio de captura genera un pulso de disparo a la entrada de trigger del *ADC* y se inicia la captura de las muestras y el barrido de la escena. El motor de elevación, tras recibir el mismo pulso que el *ADC*, inicia un movimiento desplazamiento lineal del husillo que se convierte en una velocidad angular casi constante sobre γ (véase 3.2.3). La elipse se transforma entonces en una curva abierta por cada rotación completa en ϕ .

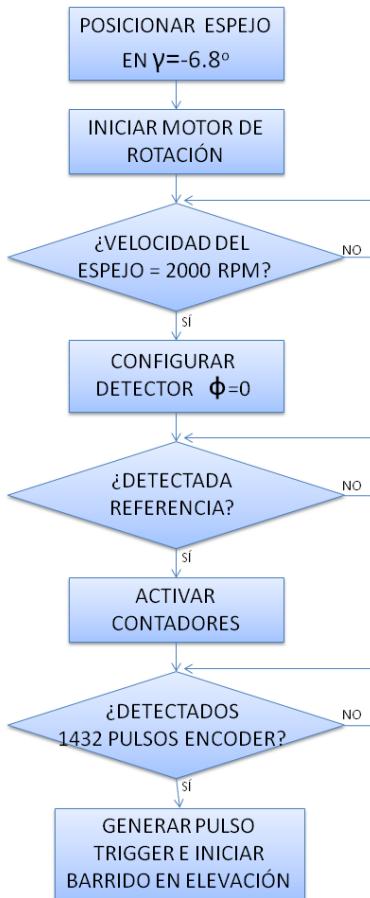


Figura 3.31: Diagrama de flujo del sistema de sincronismo.

La figura 3.32 muestra una simulación de un barrido completo de la escena mediante 39 rotaciones completas en ϕ y una variación en γ de -6.8 a $+6.8$ grados. La velocidad de rotación del espejo respecto de su *eje de rotación* (véase 3.1.4) es de 2,000 rpm. La velocidad de rotación del espejo respecto de su *eje de elevación* es de tan solo 1.88 rpm que equivalen a $11.33 \frac{\circ}{s}$, siendo ésta una velocidad media (véase 3.2.3). Por ello, al *eje de rotación* también se le denomina **eje de rotación rápido** y al *eje de elevación* se le denomina **eje de rotación lento**. La figura ha sido generada en Matlab y su código está disponible en el anexo F.1.

Una vez almacenado el número de muestras necesario para analizar la escena se detienen ambos motores y se recupera la orientación inicial $\gamma = -6.8$ grados.

El tiempo total invertido en el barrido de la escena con estos parámetros de rotación se muestra en la ecuación 3.8. A este tiempo hay que sumarle el tiempo que tarda el motor en alcanzar la velocidad de trabajo si éste parte del reposo. Para una velocidad de rotación del espejo de 2,000 rpm el tiempo invertido en tal cometido es 0.9 segundos.

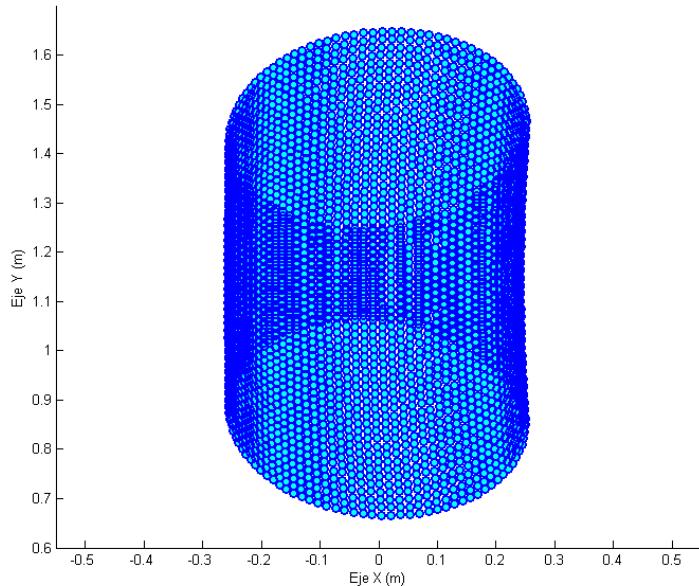


Figura 3.32: Trayectoria descrita por el *spot* en el plano del objetivo.

$$t_{captura} = \frac{6.8^\circ \cdot 2}{11.33 \frac{^\circ}{s}} = 1.2 \text{ segundos.} \quad (3.8)$$

3.2. Subsistema de apuntamiento.

La dirección de apuntamiento del sistema de antenas depende de la orientación del *espejo de exploración* (véase el apartado 3.1.4). Dos motores son los encargados de modificar la orientación. Uno de ellos se denomina motor de rotación y establece el ángulo de rotación del espejo respecto del *eje de rotación*. Dicho ángulo es denominado ángulo ϕ (véase el apartado 3.1.4). El otro se denomina motor de elevación y establece el ángulo de rotación del espejo respecto del *eje de elevación*. Este otro ángulo se denomina ángulo γ .

En la figura 3.33 se muestra un corte transversal de la estructura que sustenta ambos motores junto con el *espejo de exploración* representado en color azul claro. El motor de rotación se ha representado en color gris mientras que el motor de elevación se ha representado en color rojo. En dicha figura el ángulo γ tiene el valor de 0 grados y, por lo tanto, el *eje de rotación* se ubica en una línea horizontal imaginaria que atraviesa el centro del espejo.

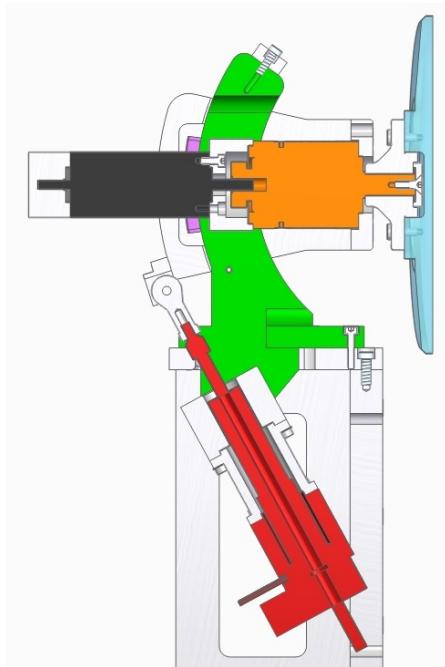


Figura 3.33: Estructura soporte del espejo de exploración.

3.2.1. Simulación del subsistema de apuntamiento.

Durante la fase de diseño de las antenas, el equipo de la Universidad de Vigo ha simulado mediante la aplicación *GO ray tracing* un trazado de rayos para determinar la posición de apuntamiento en el *plano focal* (véase la sección 3.1.3) en función de los ángulos γ y ϕ (distíngase *plano focal* de *plano del objetivo*).

La dirección de apuntamiento de la antena depende del vector normal a la superficie del espejo de exploración. Dicho vector normal depende a su vez de los ángulos γ y ϕ (véase el apartado 3.1.4). Al margen del movimiento descrito por el sistema de antenas (véase el apartado 3.1.9), para todo punto del *plano focal*, existe al menos una combinación de γ y ϕ que orienta adecuadamente el espejo y permite realizar dicho apuntamiento.

El vector normal a la superficie del espejo para una orientación determinada según γ y ϕ tiene la dirección de apuntamiento mostrada en la ecuación 3.9, donde \hat{x} e \hat{y} son respectivamente los vectores unitarios de los ejes X e Y mostrados en la figura 3.9.

$$\vec{n} = -[\sin\beta \cos\phi] \hat{x} - [\sin\gamma \cos\beta + \cos\gamma \sin\beta \sin\phi] \hat{y} + [\cos\gamma \cos\beta + \sin\gamma \sin\beta \sin\phi] \hat{z} \quad (3.9)$$

Las componentes \hat{x} e \hat{y} se denominan u y v respectivamente según se muestra en las ecuaciones 3.10 y 3.11).

$$u = -\sin\beta \cos\phi \quad (3.10)$$

$$v = -\sin\gamma \cos\beta - \cos\gamma \sin\beta \sin\phi \quad (3.11)$$

La figura 3.34 muestra en trazos de color negro los puntos analizados el plano del objetivo. Cada trazo vertical se corresponde con un valor constante de la componente u y una variación de la componente v comprendida entre los valores [-0.2, 0.2].

Cada trazo horizontal se corresponde con un valor constante de la componente v y una variación de la componente u comprendida entre los valores [-0.1, 0.1].

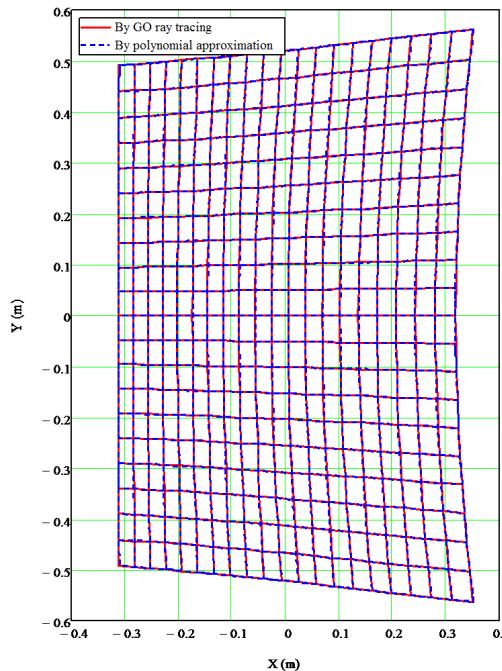


Figura 3.34: Apuntamiento de rayos en el plano focal.

La función bidimensional (x,y) descrita en la figura se puede aproximar mediante 2 polinomios (ecuaciones 3.12 y 3.13) que modelan las coordenadas X e Y del plano focal respecto del sistema de coordenadas definido en 3.9. Dicha aproximación se representa en color rojo en la misma figura.

$$X_F = a_0 + a_1 u + a_2 u^2 + a_3 v^2 + a_4 u^3 + a_5 u v^2 \quad (3.12)$$

$$Y_F = b_1 v + b_2 u v + b_3 u^2 v + b_4 v^3 \quad (3.13)$$

La obtención de las ecuaciones que modelan el apuntamiento de la antena no forma parte del alcance de este proyecto.

Parámetro	Descripción	Valor
a_0	Término independiente	-0.02586 m
a_1	Término lineal de u	-3.08996 m
a_2	Término cuadrático de u	2.89638 m
a_3	Término cuadrático de v	0.43756 m
a_4	Término cúbico de u	-6.60288 m
a_5	Término de tercer orden de uv^2	-3.85521 m
b_1	Término lineal de v	-2.51789 m
b_2	Término de segundo orden de uv	1.77152 m
b_3	Término de tercer orden de u^2v	-3.76793 m
b_4	Término cúbico de v	-2.07521 m

Tabla 3.5: Valor de las constantes de las ecuaciones 3.12 y 3.13.

3.2.2. Motor de rotación.

El subsistema de rotación se compone del motor de rotación, un encoder que facilita su posición angular, una reductora y un controlador que excita el motor. En la figura 3.35 se muestra las conexiones existentes entre estos dispositivos.

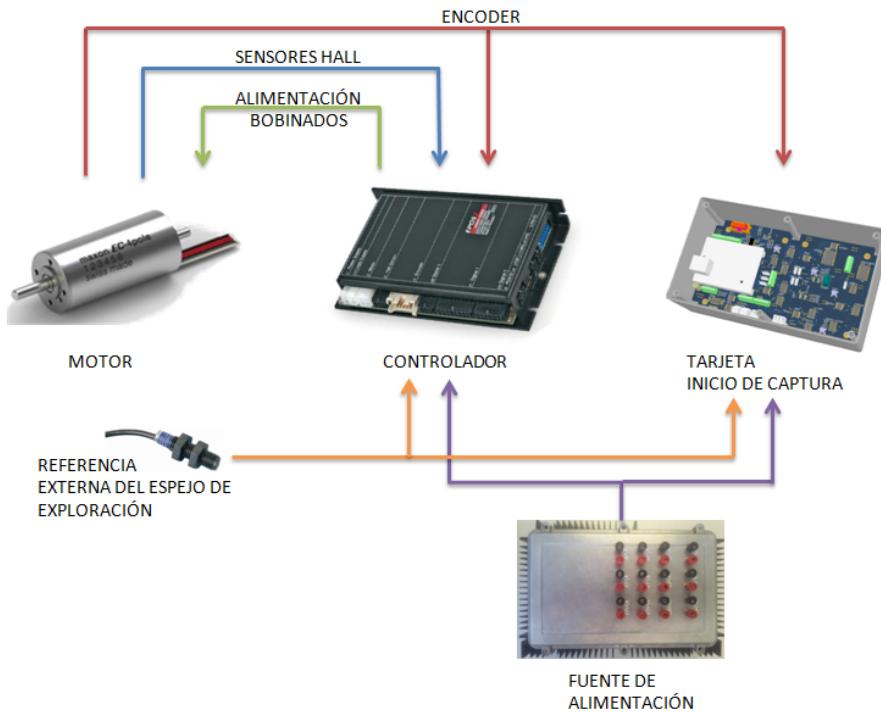


Figura 3.35: Conexiones entre los elementos del sistema de rotación.

El motor encargado de la rotación del espejo es de la marca Maxon y modelo EC-4 Pole 305013. Tiene una potencia de 200W y es capaz de girar hasta 16,700 rpm aunque su velocidad nominal (aquella velocidad en la cual se maximiza el par) es de 15,900 rpm. Su estator está compuesto por 2 bobinados de 24V que generan 2 pares de polos. Dichos

bobinados soportan una corriente nominal de 9.65 amperios y tienen una constante térmica de tiempo de 2.11 segundos. En la tabla 3.6 se detallan más características técnicas.

Parámetro	Valor
Resistencia en terminales	0.102 Ω
Inductancia en terminales	0.016 mH
Constante de par	13.6 mNm/A
Constante de velocidad	700 rpm
Relación velocidad/par	5.21 rpm/mNm
Constante de tiempo mecánica	1.82 ms
Momento de inercia del motor	33.3 gcm^2

Tabla 3.6: Características del motor de rotación.

El par nominal es de 124 mNm y soporta un par de arranque de 3,220 mNm que impone una corriente máxima en los bobinados de 236 amperios. Dispone en su interior de 3 sensores hall que facilitan una posición aproximada del rotor. Esta información es imprescindible para que el controlador (véase el apartado 3.2.2.3) pueda iniciar el movimiento del motor adecuadamente (el controlador necesita conocer cuál es el bobinado que produce el giro en el sentido correcto cuando éste es excitado).

3.2.2.1. Reductora.

El acoplamiento entre el espejo de rotación y el motor se realiza a través de una reductora 4:1 (figura 3.36) de la marca Neugart y modelo PLE40 que consigue que por cada 4 vueltas completas de giro del motor el espejo de una. De esta forma se consigue aumentar el par en el eje de rotación del espejo a costa de disminuir la velocidad de giro de éste. El par máximo soportado a la salida de la reductora es de 15 Nm.



Figura 3.36: Reductora PLE 40-4.

En la figura 3.33 se muestra un corte transversal de la estructura que sustenta los motores donde aparece representada la reductora mediante el color naranja. La reductora

presenta una holgura inferior a 15 arcmin (0.25 grados) en el acople del motor a la misma que ha de ser tenida en cuenta en los cambios de signo de la aceleración aplicada al motor.

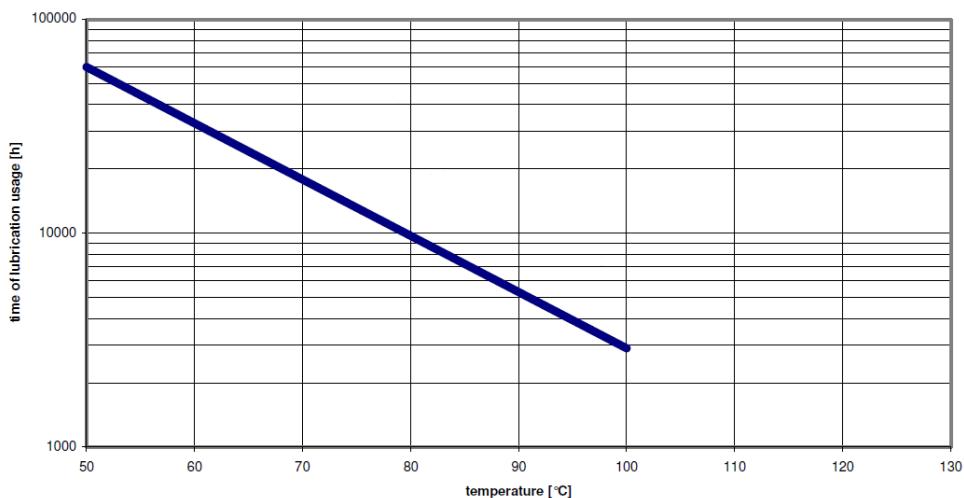


Figura 3.37: Vida útil del lubricante de la reductora.

La máxima velocidad de giro soportada a la entrada es de 18,000 rpm. Tiene una eficiencia del 96 % y genera un nivel de ruido de 58 dB(A). En su interior dispone de un lubricante cuya vida útil depende de la temperatura de trabajo. La vida útil de la reductora queda determinada por la vida útil del lubricante ya que éste no se puede sustituir. Tras 10 minutos de funcionamiento continuo a una temperatura ambiente de 21 grados y con el motor girando a 8,000 rpm, la temperatura de la carcasa de la reductora alcanza los 51 grados. Observando la figura 3.37 se comprueba que en estas condiciones la vida útil del lubricante de 60,000 horas.

3.2.2.2. Encoder.

El motor tiene acoplado un encoder con referencia de Maxon 225780 (figura 3.38) que permite controlar el ángulo de giro del motor. El encoder es relativo y, por tanto, es necesaria una referencia externa adicional para poder conocer el ángulo de rotación respecto de dicha referencia.

El encoder dispone de una señal, denominada Index, que genera un pulso por cada vuelta de motor y de dos señales en cuadratura denominadas canal A y canal B. Las señales se transmiten en modo diferencial para minimizar el efecto del ruido inducido desde el encoder hasta la electrónica que gestiona el motor. Las señales se convierten a modo común mediante un receptor de linea. Los canales A y B generan cada uno de ellos 1,000 pulsos por cada vuelta de motor. El canal A presenta un desfase de 90 grados respecto del canal B. Mediante el circuito apropiado se consiguen generar 4 pulsos por



Figura 3.38: Encoder del motor de rotación.

cada pulso del canal A. Cada uno de estos pulsos se denomina qc (Quad Count) que a su vez son las unidades de posicionamiento que utiliza el controlador del motor (3.2.2.3). En una vuelta del *espejo de exploración* se detectan 16,000 qc's.

El encoder permite una velocidad de giro máxima de 12,000 rpm limitando por tanto hasta este valor la velocidad de giro del motor. Dispone de una cinta plana de paso 2.54 mm y de 16 vías terminada en un conector IDC hembra de 16 posiciones. A través de dicho conector el encoder facilita todas las señales y proporciona un par de cables para su polarización. La cinta plana está conectada en paralelo con el controlador del motor (véase 3.2.2.3) y con la *tarjeta de inicio de captura* (descrita en el apartado 5.1.3). La polarización requerida por el encoder es de 5 V; Esta alimentación es facilitada por el controlador.

3.2.2.3. Controlador.

El motor y el encoder están conectados a un controlador modelo EPOS2 70/10 (figura 3.39). El controlador excita sinusoidalmente los bobinados del motor con la ayuda de los sensores hall y del encoder. De esta forma, el controlador es capaz de gestionar la velocidad de giro del motor, su aceleración y su posicionamiento. Además, vigila que la potencia aplicada no dañe el motor. Para ello el controlador tiene en cuenta la máxima corriente de pico que soportan los bobinados y la constante de tiempo térmica de los mismos.

El controlador permite un rango de entrada de alimentación comprendida entre 11 y 70V de corriente continua. Si la potencia disponible a la entrada no es suficiente para satisfacer las necesidades demandadas en tiempo real para un determinado régimen de funcionamiento del motor, el controlador dispara una alarma mediante un indicativo luminoso y deja de excitar los bobinados del motor (véase A.2.5)

El controlador polariza tanto los sensores Hall como el encoder mediante una tensión de 5 V. Dispone de un puerto USB 2.0 (12 Mbps) para comunicarse con un PC y poder configurar el motor con la ayuda de la GUI (Graphic User Interface) que facilita el propio fabricante. La descripción de éste software se adjunta en el anexo A.2



Figura 3.39: Controlador del motor de rotación.

El software del controlador contiene un *"diccionario de objetos"* (figura 3.40) en el que se integran todos los parámetros, estados y acciones que soporta el motor. Cada objeto del diccionario tiene asociada una dirección única de memoria (organizada como *index* y *subindex*) y un tamaño de bytes determinado. Por ejemplo, la versión firmware del controlador se corresponde con el objeto del diccionario cuyo *index* es 0x2003 y cuyo *subindex* es 0x01. En este caso el objeto es de solo lectura y, por tanto, no se permite su modificación por parte del usuario.

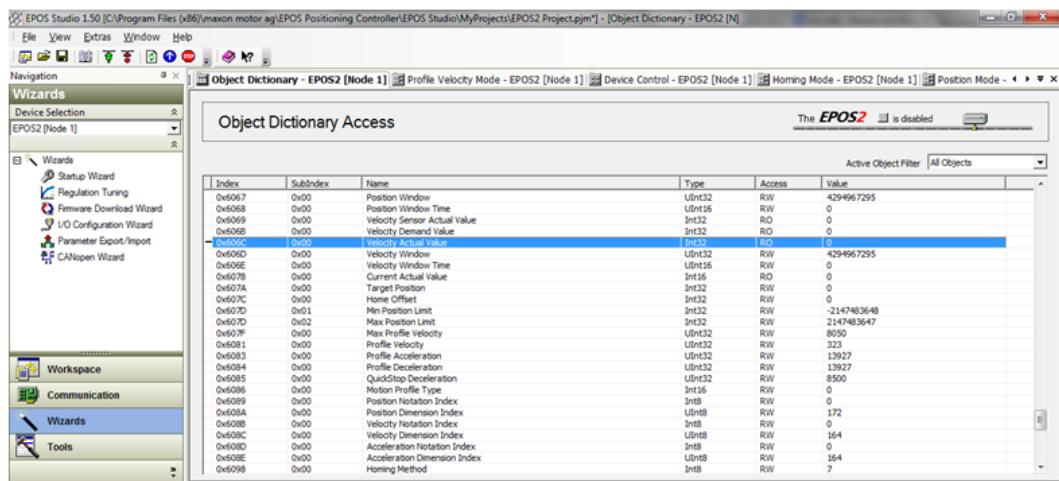


Figura 3.40: Diccionario de objetos del controlador EPOS2 70/10.

También cuenta con un bus *CANopen* (estándar de comunicaciones industrial) capaz de transferir datos a 1 Mbps y con un puerto serie *RS232* cuya velocidad máxima es de 115 Kbps. Mediante estos interfaces también es posible consultar o modificar cualquier objeto del diccionario de objetos.

La transferencia de datos utiliza técnicas *ARQ* (*Automatic Repeat Request*) para la confirmación de la recepción de los mensajes y en estos últimos se incluyen

códigos *CRC* (*Cyclic Redundant Check*) para detectar posibles errores en la transmisión.

El fabricante pone a disposición del desarrollador una *DLL* (Dynamic Library Link) para controlar la unidad mediante programación en Visual C. Para ello utiliza una comunicación serie a través de un objeto *COMM* el cual hace uso de la interfaz USB. En el anexo A.4 se detalla el funcionamiento de dicha librería, su instalación y sus principales funciones.

El controlador es capaz de gestionar el motor de tres formas diferentes:

- Mediante un control de la corriente aplicada al estator.
- Mediante un control de la velocidad del motor.
- Mediante un control de la posición del rotor.

La descripción de los diferentes modos de regulación está disponible en el anexo A.1. El radar funciona actualmente con el controlador en modo control de velocidad. Dicho controlador es configurado por la aplicación del sistema (descrita en el apartado 5.2) a través de la DLL correspondiente.

Para compensar los efectos inerciales de la mecánica acoplada al eje del rotor el controlador incorpora un asistente software denominado *Regulation Tuning*. Dicho asistente realiza diversas pruebas de movimiento, de control de corriente y de control de velocidad para ajustar los parámetros que mejor se ajusten a la mecánica acoplada de forma que el motor responda de la forma óptima a las demandas impuestas por el controlador. Los detalles de la calibración del motor se encuentran disponibles en el anexo A.3.

3.2.2.4. Caracterización del motor.

Se han realizado pruebas de rotación para poder determinar la máxima velocidad del motor tolerada en el sistema. El resultado de las mismas indican que, para la estructura a la que está acoplado el motor, la mayor velocidad permisible es de 9,000 rpm. A mayores velocidades se perciben fuertes vibraciones en la estructura y el controlador no es capaz de fijar correctamente la velocidad (véase 3.1.4).

Por otro lado, se ha capturado con una tarjeta *ADC* los pulsos correspondientes al canal A (véase 3.2.2.2) generados por el encoder cuando el motor gira a una velocidad constante de 8,000 rpm para comprobar la estabilidad de la velocidad de giro. En las medidas realizadas el controlador no aplica el filtro descrito en el apéndice A.1.2, no obstante, no se han apreciado variaciones de consideración al repetir las mismas con el filtrado activado.

El *ADC* ha codificado con 16 bits la señal a razón de 40 Megamuestras por segundo capturando en total 200 Megamuestras que equivalen a 166 rotaciones completas

del espejo. Posteriormente y con la ayuda de Matlab se han procesado las muestras capturadas.

Lo primero que se ha tenido que realizar en el procesado es un filtrado de la señal ya que ésta presenta ruido procedente de los sensores del propio encoder que dificulta la detección de la duración de cada pulso. En la figura 3.41 se muestra la señal del encoder capturada y la misma tras el filtrado.

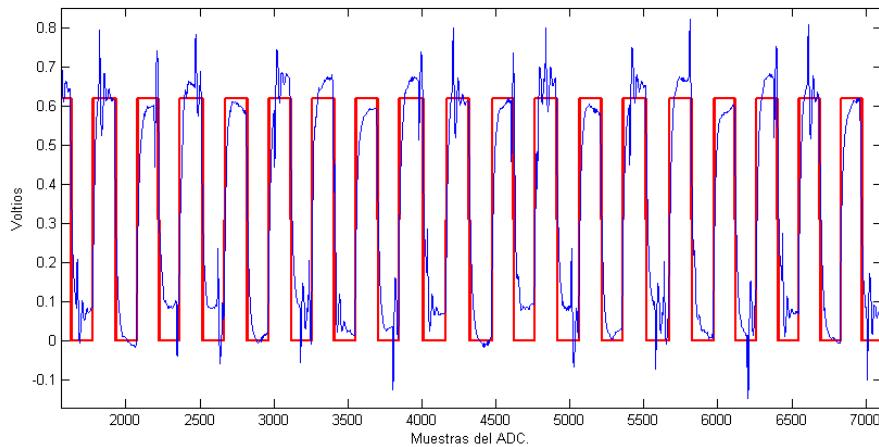


Figura 3.41: Filtrado de los pulsos del encoder.

En la figura 3.42 se presenta la función densidad de probabilidad de la velocidad instantánea (entendiéndose por velocidad instantánea aquella que se calcula tomando como referencia dos pulsos consecutivos del canal A).

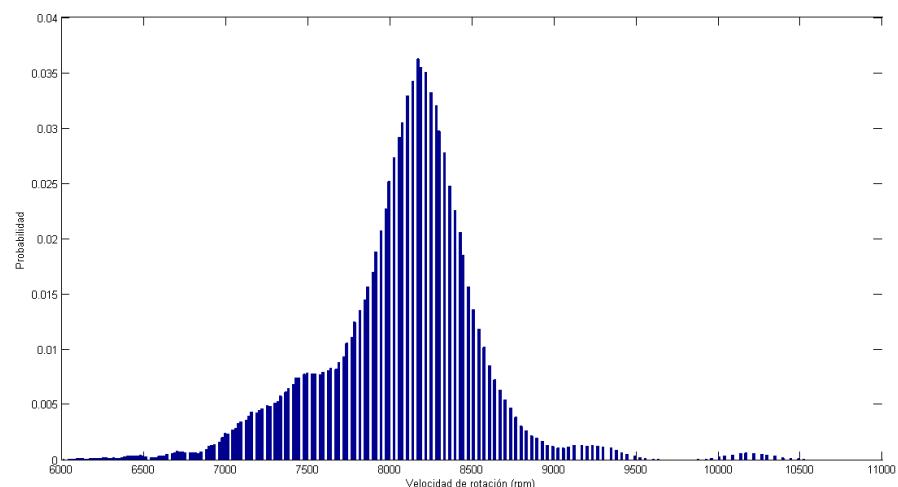


Figura 3.42: Función densidad de probabilidad de la velocidad instantánea.

Para calcular la velocidad media de cada vuelta se calcula el tiempo transcurrido en detectar los 4,000 pulsos del canal A correspondientes a una vuelta del *espejo de exploración*. En la figura 3.43 se representa la evolución de la velocidad a lo largo de las 166 vueltas capturadas.

El resultado de la caracterización confirma que el controlador del motor no es capaz de hacer que el motor mantenga una velocidad constante. La causa de esta incapacidad se debe a que el espejo de exploración que el motor tiene acoplado no tiene su centro de masas sobre el eje de éste (véase el apartado 3.1.4).

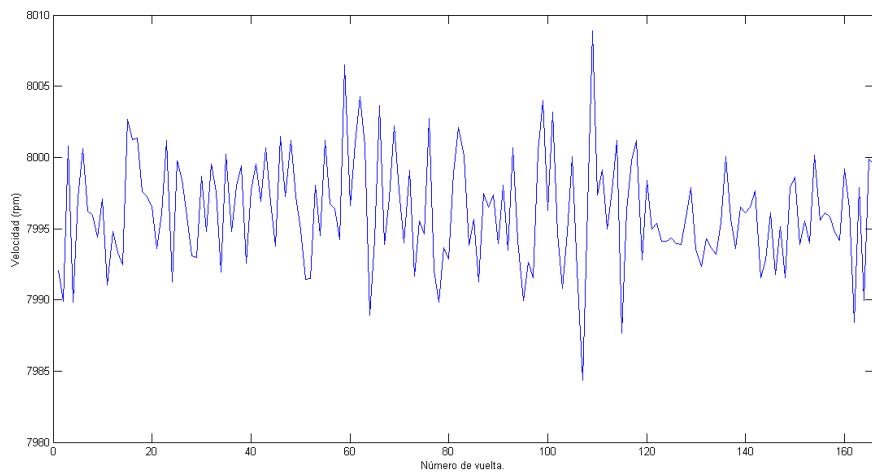


Figura 3.43: Velocidad media de rotación.

3.2.3. Motor de elevación.

El motor de elevación es el encargado de modificar el ángulo de elevación γ (véase 3.1.4). Se procede a realizar una somera descripción del funcionamiento del motor para, posteriormente, realizar una descripción formal en los respectivos apartados. El motor de elevación extiende una varilla metálica denominada **husillo** que está conectada a la pieza móvil que sustenta el motor de rotación. De tal forma, la extensión del *husillo* modifica la inclinación del *eje de rotación*. El motor de elevación produce un giro de 90 grados en su rotor cada vez que su controlador le manda un pulso. Dicho giro se denomina **step**. Cada *step* se corresponde con una extensión determinada del *husillo*. De tal forma, el *husillo* se extiende con un paso milimétrico fijo que incrementa o disminuye la distancia "d" de la figura 3.44 la cual es un detalle de la figura 3.33.

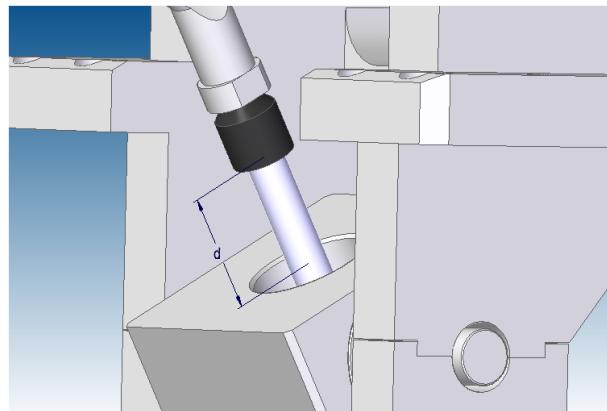


Figura 3.44: Movimiento angular de elevación.

A pesar de que cada *step* del motor se traduce en un incremento constante en el desplazamiento del *husillo*, cada incremento de éste último (incremento de la distancia *d* de la figura 3.44) se traduce en una variación del ángulo gamma cuyo incremento depende del valor de γ según se muestra en la figura 3.45. El controlador, a pesar de tener capacidad para poder definir una aceleración y una deceleración en el movimiento no es capaz de corregir este fenómeno (la resolución de la aceleración no es suficientemente precisa).

El código desarrollado para determinar la posición de apuntamiento del sistema de antenas aproxima la velocidad angular de la figura 3.45 en tres tramos lineales (véase el apartado 3.2.4.5).

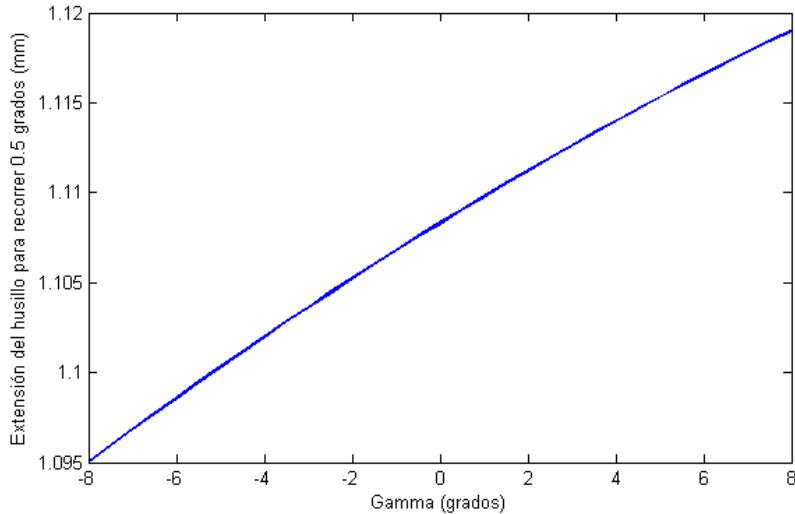


Figura 3.45: Movimiento angular de elevación.

El motor es capaz de modificar el ángulo γ hasta los límites que impone la estructura, los cuales se corresponden con valores de $\gamma = \pm 10.1$ grados. El ángulo de elevación γ no varía de forma constante con el tiempo cuando el motor de elevación trabaja a velocidad constante. En la figura 3.46 se muestran los elementos que intervienen en el subsistema de elevación.

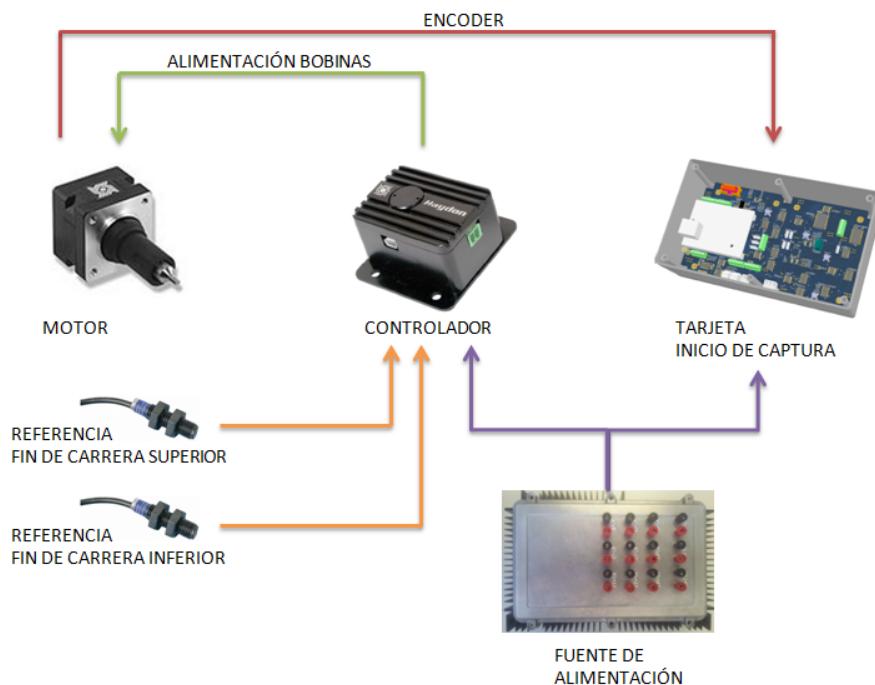


Figura 3.46: Subsistema de elevación.

El motor de elevación (figura 3.47) es un motor lineal de pasos captivo (el motor se queda fijo y lo que se desplaza el *husillo*) de la marca Haydon y modelo 35H4B-2.33-019.



Figura 3.47: Motor de elevación.

Un motor lineal se diferencia de un motor de rotación clásico porque en lugar producirse un torque, o fuerza de giro, cuando se aplica una corriente al estator lo que se genera es una fuerza lineal que provoca un desplazamiento lineal de un eje. El hecho de que el motor sea de pasos implica que el movimiento lineal no es continuo sino discreto. Para ello el motor en cuestión dispone de dos bobinados en su estator (estator de 4 polos) denominados bobinados A y B que son excitados mediante pulsos de corriente.

La orientación del rotor dependerá de por cuál de los bobinados A o B circule corriente como se indica en la siguiente figura:

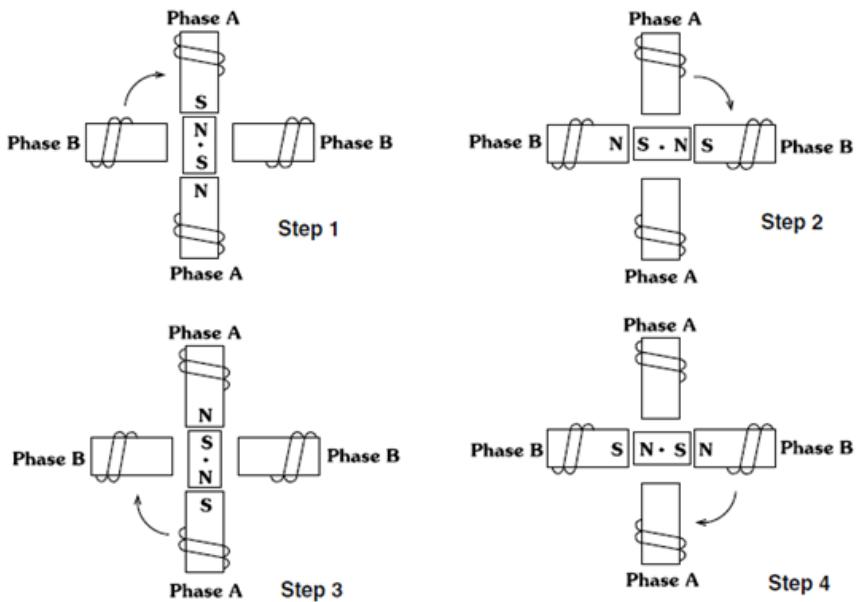


Figura 3.48: Motor por pasos.

Se denomina *step* o paso a cada 90 grados de giro del rotor. Si en lugar de polarizar los bobinados alternativamente en cada *step* se polarizan los mismos de forma simultánea pero alternando el sentido de la corriente [figura 3.49] se logra un 41 % más de torque en el rotor.

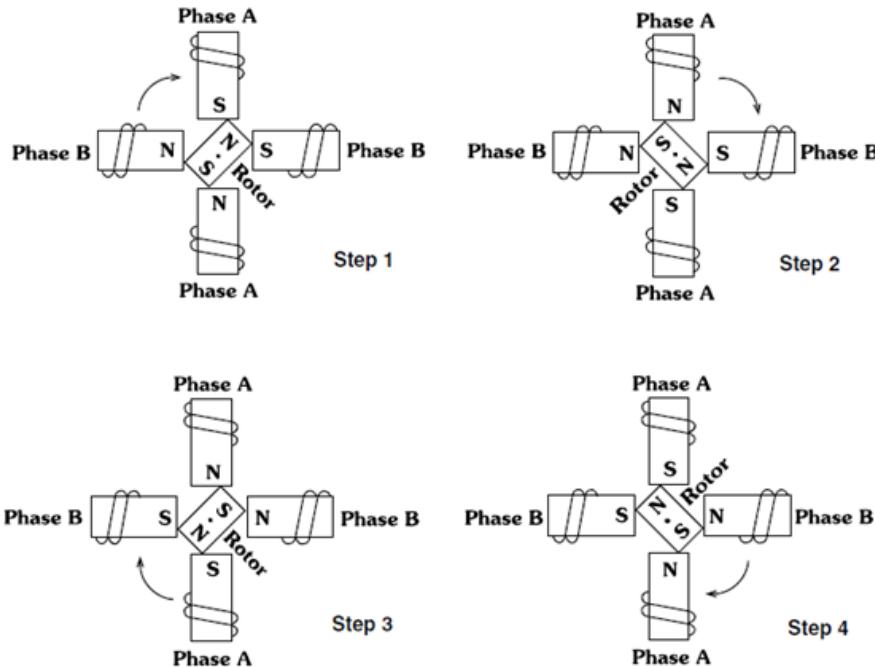


Figura 3.49: Motor por pasos con polarización alternada.

Si se combinan ambas estrategias se puede obtener giros discretizados del rotor de 45 grados o $\frac{1}{2} step$. Mediante técnicas más avanzadas de desfase se logran obtener movimientos más precisos.

El rotor está acoplado a un *husillo* como se indica en la figura 3.50 donde se ha representado al rotor mediante una tuerca. Puesto que el motor es de tipo captivo éste permanecerá fijo durante el movimiento. Al girar el rotor lo que se produce es un desplazamiento del *husillo* puesto que éste no puede rotar sobre su eje propio por no tratarse una pieza con simetría de revolución

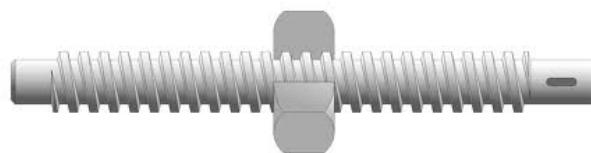


Figura 3.50: Husillo del motor de elevación.

Se disponen de 2 versiones del motor de elevación como opción; versión B y C. Ambos tienen bobinados de 2.33V. La diferencia estriba en el avance del eje por cada *step* aplicado. En la versión B el eje se prolonga 0.0158 mm por cada *step* mientras que en la versión C la prolongación es justo el doble (0.0317 mm/*step*). En la figura 3.51 se muestran las curvas de carga que soportan ambos motores para una velocidad dada expresada en $\frac{steps}{s}$.

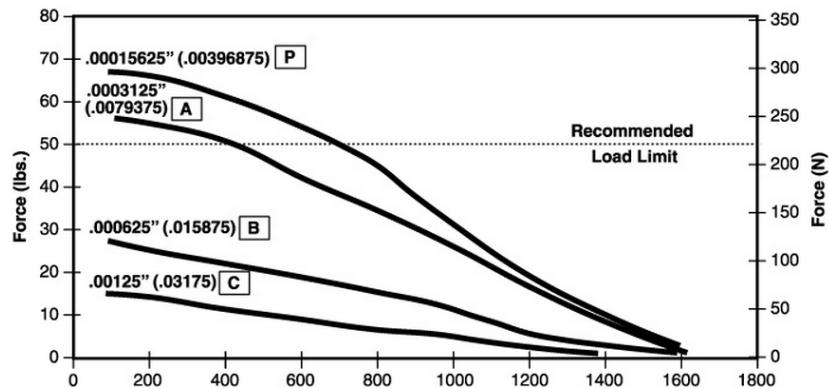


Figura 3.51: Curvas de carga de los motores de elevación.

Se ha comprobado que ambos motores son capaces de desplazar la estructura acoplada a razón de 1,600 *steps* por segundo. Actualmente el radar utiliza la versión C.

3.2.3.1. Encoder.

El motor de elevación dispone de un encoder relativo de 1,000 pulsos similar al encoder descrito en el apartado 3.2.2.2. También se polariza a 5V y transmite su información mediante 3 señales diferenciales denominadas Index, canal A y canal B. Sin embargo, el controlador no permite la conexión de este tipo de encoder. Actualmente la información de este encoder no está siendo utilizada por el sistema de control y sincronismo del sistema.

3.2.3.2. Controlador.

El motor de elevación se conecta al controlador Haydon modelo PCM4806X (figura 3.52). Dicho controlador admite una polarización comprendida entre los 12 y 48V de corriente continua.



Figura 3.52: Controlador del motor de elevación.

Dispone de un conector de entradas y salidas optoaisladas con un rango de voltajes tolerados comprendidos entre 5 y 24V, de un puerto de serie RS485 para la comunicación con otros dispositivos y de un puerto USB 2.0 para comunicarse con un PC. Con el

controlador se incluye un conversor de RS485 a RS232 pudiéndose de esta manera elegir el tipo de interfaz que más interese.

El controlador dispone de un software (*IDEA interface program*) para interactuar con el motor e implementar programas en el mismo. Dicho software está disponible en la web del fabricante (<http://www.haydonkerk.com>).

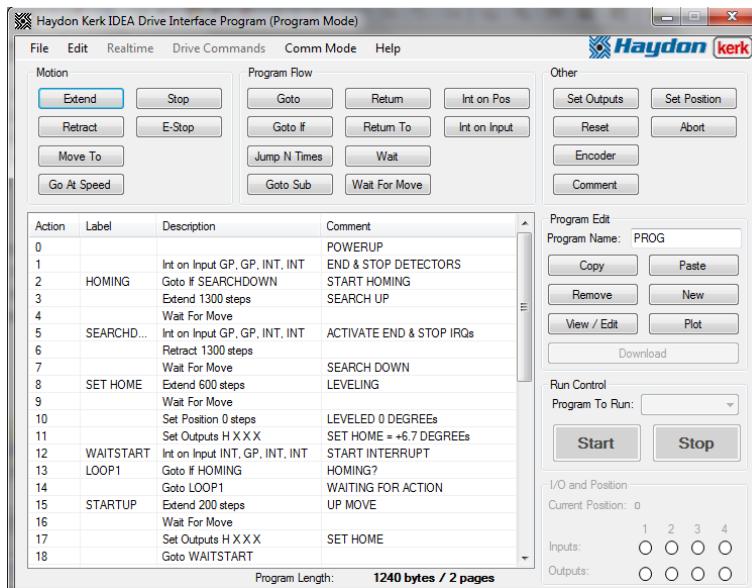


Figura 3.53: GUI del controlador Haydon.

El controlador dispone de una memoria interna capaz de albergar distintos programas. También se le puede indicar al controlador que ejecute un programa por defecto cuando éste es polarizado.

La *GUI* también permite manejar el controlador en tiempo real. Esto sucede si se pulsa en el comando *Realtime*. En este modo, en lugar de insertarse una línea en el programa en curso cuando se pulsa una acción, lo que se hace es transmitir dicha orden al controlador.

El controlador permite realizar movimientos con una aceleración y deceleración determinadas y con una resolución del avance que puede variar desde un *step* hasta 1/64 *step* (figura 3.54).

Otra forma de interactuar con el controlador al margen de la *GUI* es mediante comandos a través del puerto COM virtual. Mediante un terminal de comunicaciones cualquiera y una conexión de 57600 bps, 8 bits de datos, un bit de inicio, otro de parada y ninguno destinado al control de la paridad, se le puede enviar comandos al controlador.

Existen comandos de ejecución en tiempo real con acciones determinadas como extender o retraer el *husillo*, comandos para modificar el comportamiento de los programas que se alojan en memoria, y, comandos para transferir nuevos programas o borrarlos de la memoria.

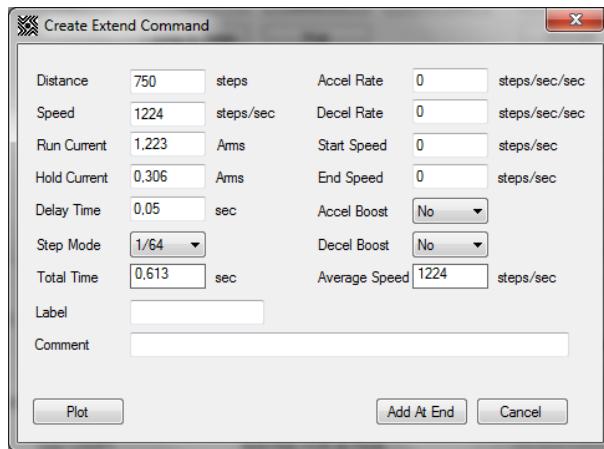


Figura 3.54: Maniobra de extensión.

Así, por ejemplo, el comando " $I-1200,3200,40000,100000,1600,500,1900,2000,50,8$ " genera un movimiento inmediato de extensión de 1200 *steps* a una velocidad de 3,200 $\frac{steps}{s}$ con una aceleración de 40,000 $\frac{steps}{s^2}$, una deceleración de 100,000 $\frac{steps}{s^2}$ con una corriente de excitación de los estatores de 1.6 amperios durante el movimiento a velocidad constante, una corriente de 0.5 amperios fija cuando finaliza el movimiento, una corriente de excitación durante la aceleración de 1.9 amperios y una corriente de excitación durante la deceleración de 2.0 amperios. El penúltimo parámetro indica el tiempo transcurrido entre el último *step* aplicado en el movimiento y el inicio de la excitación (0.5 segundos a 0.5 amperios) para mantener la posición. El último parámetro indica las unidades de avance. En este caso concreto se trata de un movimiento que tiene una resolución de $\frac{1}{8}$ de *step*.

3.2.4. Referencias del subsistema de apuntamiento.

Los motores del sistema disponen de sendos encoders relativos (véase 3.2.2.2 y 3.2.3.1). Los encoders relativos generan un pulso cuando el rotor incrementa su posición angular un ángulo θ equivalente a la resolución del encoder. Sin embargo, los encoders relativos no son capaces de facilitar la posición angular del rotor respecto de un sistema de coordenadas fijo. Para lograr obtener dicha información es necesario determinar un origen angular respecto de un sistema fijo y posteriormente medir, mediante los pulsos generados por el encoder, el incremento angular recorrido desde dicho origen.

El origen angular se determina mediante un detector y una referencia externa adherida a la parte móvil. La referencia externa excita el detector de la parte móvil en una posición determinada de la misma.

3.2.4.1. Referencia del motor de rotación.

El dispositivo utilizado para detectar la referencia externa y poder así establecer un origen $\phi = 0$ es un sensor inductivo de proximidad de la marca Telemecanique y modelo XS4P08NA340, el cual se muestra en la figura 3.55.



Figura 3.55: Sensor inductivo.

El sensor posee un devanado de cobre por el que circula una corriente que genera un campo magnético. Al acercarse un cuerpo metálico al devanado se inducen en éste, por efecto de dicho campo, corrientes de *Foucault* que a su vez generan un campo que contrarresta el campo generado por el sensor. La variación del campo se manifiesta en una variación de la impedancia del devanado, la cual es analizada por un circuito detector que se integra dentro del mismo sensor.

El sensor es capaz de detectar un objeto metálico enfrentado a menos de 2.5 mm de distancia respecto de su superficie de detección. Cuando esta situación sucede un led del propio sensor se ilumina y se conmuta el nivel lógico de su salida digital.

La frecuencia máxima que soporta el sensor es de 5 KHz y su máxima corriente de conducción es de 200 mA. Permite un rango de alimentación comprendido entre los 12 y 24 Vdc.

Está fijado a la estructura móvil que soporta el *espejo de exploración* mediante un roscado de métrica de 8 mm según se muestra en la figura 3.56.

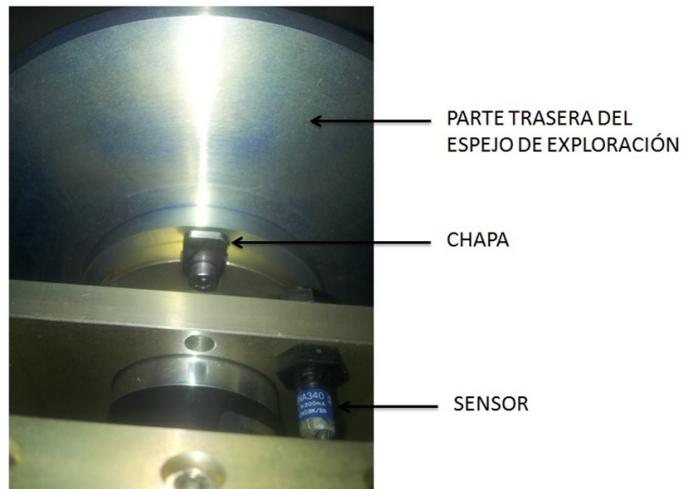


Figura 3.56: Referencia externa para la determinación de ϕ .

Una chapa, instalada en uno de los tornillos que sujetan el *espejo de exploración* a modo de referencia (mostrada en la figura 3.56), se enfrenta al detector cuando el espejo tiene un ángulo de rotación ϕ determinado excitando su salida.

La salida digital del sensor está conectada a una de las entradas digitales del controlador del motor de rotación (véase la sección 3.2.2.3) para que éste sea capaz de realizar la función *homming* (descrita en el apéndice A.2.4). El sensor también está conectado en paralelo al sistema de sincronismo diseñado (véase 5.1). El pulso generado por el circuito detector excita un báscula tipo *JK* que activa la señal *enable* de los contadores implementados en dicho sistema de sincronismo. Los pulsos del encoder del motor de rotación proporcionan la señal de reloj a dichos contadores una vez que éstos son activados.

3.2.4.2. Referencias del motor de elevación.

El motor de elevación es capaz modificar la inclinación del eje de rotación mas allá de los límites mecánicos que impone la estructura (véase 3.2.3). Para no dañar la mecánica ni el motor se han instalado dos sensores iguales al descrito en 3.2.4.1 y sus respectivas referencias externas que sirven al controlador del motor de elevación (véase 3.2.3) como detectores de fin de carrera.

Las referencias que excitan los sensores son dos chapas que están acopladas a la estructura fija del espejo de exploración. En la figura 3.57 se muestra en color amarillo la estructura fija y en color rojo ambas referencias. Los sensores están roscados a la parte móvil de la estructura que sustenta el motor de rotación según se muestra en la figura 3.58.

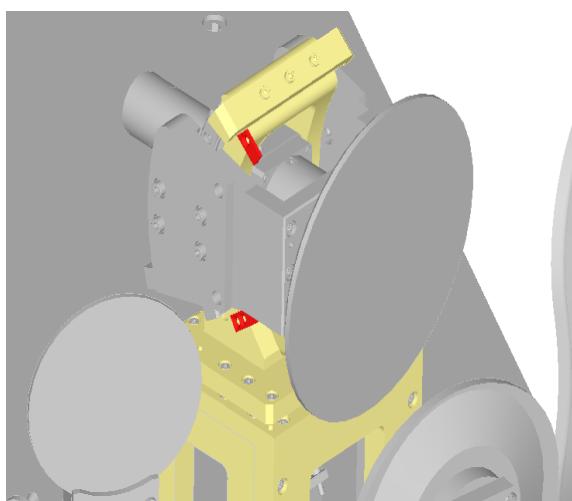


Figura 3.57: Chapas de los fines de carrera.



Figura 3.58: Sensores inductivos.

El sensor inferior sirve como referencia para poder determinar el ángulo γ . Una vez detectada la referencia el controlador inicia un movimiento de extensión del *husillo* de

827 *steps* (versión C del motor) que posiciona el eje de rotación con un valor γ de 0 grados.

El controlador desconoce el valor real de γ . Si el motor aplica cada *step* correctamente sin sufrir ningún bloqueo el controlador puede estimar la posición γ con bastante precisión. No obstante, si se produce algún bloqueo, existe algún fallo en la programación o no se excitan correctamente los estatores por fallo electrónico o de cableado puede darse el caso de que el controlador trate de realizar movimientos que la articulación mecánica no permita. Para evitar esta situación ambos sensores de fin de carrera están conectados al controlador como interrupciones externas. Si durante alguna maniobra se detecta una de estas interrupciones se detiene el movimiento para evitar daños al motor y a las piezas mecánicas.

3.2.4.3. Referencia del plano del objetivo.

Para poder ubicar con precisión objetos en el plano del objetivo se ha instalado en una de las esquinas del tablero de la mesa que soporta todo el sistema de antenas un puntero láser (véase la figura 3.59). Dicho puntero apunta a las coordenadas $(X_T, Y_T) = (0, 0)$ del plano del objetivo. El puntero utilizado es del fabricante Global Laser y modelo LaserLyte-Flex-635-5. Emite una potencia óptica de 5 mW con una longitud de onda de 635 nm. Dispone de un ajuste focal mediante el desplazamiento de una lente que permite reducir el tamaño del área iluminada en el plano del objetivo.



Figura 3.59: Puntero láser para referenciar el plano del objetivo.

3.2.4.4. Apuntamiento del sistema en el plano del objetivo.

La simulación mostrada en la figura 3.34 determina los puntos del *plano focal* que son apuntados por el sistema para un γ y ϕ determinados. Sin embargo, por tratarse de un sistema *offset*, el sistema no apunta al *plano focal* sino al *plano del objetivo* (Véase el apartado 3.1.8).

La transformación lineal mostrada en la ecuación 3.14 permite obtener la posición de apuntamiento en el *plano del objetivo* respecto del mismo sistema de coordenadas definido en la figura 3.9 y expresado en metros. El cálculo de dicha transformación y su posterior simulación en Matlab sí forman parte del alcance del proyecto.

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 8 \end{pmatrix} + \begin{pmatrix} \cos\tau & 0 & -\sin\tau \\ 0 & 1 & 0 \\ \sin\tau & 0 & \cos\tau \end{pmatrix} \begin{pmatrix} X_F \\ Y_F \\ 0 \end{pmatrix} \quad (3.14)$$

3.2.4.5. Función coordenadas del blanco.

Cuando el sistema realiza una captura de la escena (véase 3.1.9) cada muestra del *ADC* obtenida se corresponde con una orientación específica del espejo de exploración y por lo tanto con una posición de apuntamiento del sistema de antenas en el plano del objetivo.

Se ha diseñado una función en Matlab que determina las coordenadas de los blancos en el plano del objetivo en función de los números de muestra capturadas del *ADC* donde se ha determinado previamente la existencia de blancos (véase 3.1.9).

La función modela el movimiento descrito por los motores cuando ambos trabajan a velocidad constante. En dicho modelo se ha tenido en cuenta la variación angular γ descrita en el apartado 3.2.3. Para ello, se ha aproximado la curva representada en la figura 3.45 mediante tres tramos lineales de variación del ángulo γ cuyos intervalos expresados en grados son [-6.8, -1.5), [-1.5, 3) y [3, 6.8].

Una vez determinada la posición angular de γ y ϕ de cada muestra en la cual hay una detección, la función emplea las ecuaciones 3.12 y 3.12 para determinar la posición de dichos blancos en el plano del objetivo. La función incorpora los parámetros de entrada definidos en la tabla 3.7.

Parámetro	Tipo
Velocidad de rotación del espejo	Double rpm
Velocidad de barrido vertical	Double steps/s
Versión del motor de elevación	String 'B' o 'C'
Frecuencia de muestreo del <i>ADC</i>	Double Hz
Array con los # de muestra de las detecciones	Array Double adim.

Tabla 3.7: Parámetros de entrada de la función de posicionamiento de blancos.

La función retorna un array bidimensional con las coordenadas (x,y) del plano del objetivo de cada blanco. El código desarrollado está disponible en el apéndice F.1.

3.2.4.6. Representación gráfica de blancos.

Para obtener una representación gráfica de la escena capturada se ha programado una rutina en Matlab en la cual se transfiere como parámetro de entrada un array bidimensional con las coordenadas (x,y) de cada uno de los blancos detectados. El código genera con dichos datos un diagrama de dispersión en el que representa en el *plano del objetivo* cada detección.

La representación de cada blanco se realiza mediante un área circular de 1.6 cm de diámetro a pesar de que, tanto el tamaño del *spot*, como la forma del mismo, varían dependiendo de su ubicación en el plano del objetivo. El código está disponible en el apéndice F.2.

3.2.4.7. Sistema de coordenadas absolutas.

El origen de coordenadas del sistema definido en la figura 3.9 se ubica en el punto denominado *origen de antena* (véase 3.1).

No se puede definir un sistema de ejes ortogonal mediante un único punto en el espacio sin ninguna referencia adicional que ayude a determinar la orientación de, al menos, 2 de sus ejes. Por ello, se ha generado un nuevo sistema de ejes diferente al ubicado en el origen de antena ya que en este punto no es fácil definir 2 ejes. El punto elegido para tal fin es la proyección ortogonal de una de las esquinas del tablero de la mesa que sustenta el sistema de antenas sobre el suelo. En esta esquina el propio tablero determina los dos ejes necesarios. En la figura 3.60 se representa en color rojo el sistema de coordenadas definido hasta el momento y en color negro el nuevo sistema de coordenadas. El eje Y en ambos sistemas se define mediante el producto vectorial $\hat{Z} \times \hat{X}$ y $\hat{Z}_A \times \hat{X}_A$ en cada caso.

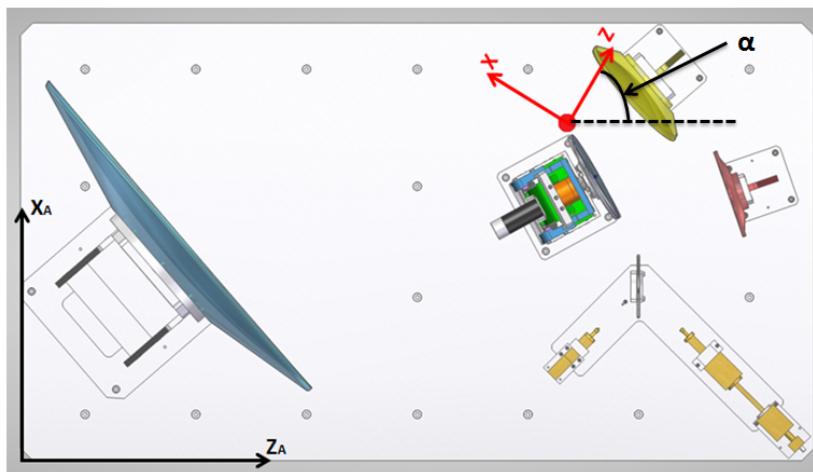


Figura 3.60: Sistema de coordenadas absolutas.

El nuevo sistema definido se denominada **sistema de coordenadas absoluto**. Dicho sistema se obtiene mediante una rotación y una traslación del sistema original.

El *origen de antena* tiene unas *coordenadas absolutas* expresadas en metros (X_A, Y_A, Z_A) = (0.569, 1.147, 0.886). Por otro lado, el eje Z forma un ángulo α de 64.58 grados con el eje Z_A .

La ecuación 3.15 muestra la transformación necesaria para obtener los puntos del *plano del objetivo* a partir de los puntos apuntados en el *plano focal* referidos en el *sistema de coordenadas absoluto*.

$$\begin{pmatrix} X_A \\ Y_A \\ Z_A \end{pmatrix} = \begin{pmatrix} 8 \cdot \sin\alpha \\ 1.147 \\ 8 \cdot \cos\alpha \end{pmatrix} + \begin{pmatrix} \cos(\alpha - \tau) & 0 & -\sin(\alpha - \tau) \\ 0 & 1 & 0 \\ \sin(\alpha - \tau) & 0 & \cos(\alpha - \tau) \end{pmatrix} \begin{pmatrix} X_F \\ Y_F \\ 0 \end{pmatrix} \quad (3.15)$$

3.3. Caracterización del sistema de antenas.

Se han realizado medidas del sistema de antenas para caracterizar la forma y tamaño del spot según la dirección de apuntamiento de la antena. Para ello, se ha ubicado en el plano del objetivo una matriz de 13 filas y 7 columnas de bolas de acero de 5 mm de diámetro. La separación existente entre filas y columnas de la matriz es de 7.5 cm.

Durante los procesos de prueba del sistema de sincronismo y apuntamiento se han podido detectar, entre otros objetivos, la presencia de hilos de nylon de 160 μm de diámetro a una distancia de 8 metros. Por tal motivo, las bolas no pueden posicionarse mediante hilos. Sobre una superficie de poliestireno expandido de 2 cm de espesor se han clavado palillos de madera de 2.5 mm de diámetro y 5 cm de longitud. En el extremo de cada palillo, se ha adherido mediante pegamento una bola de acero. Desde la ubicación del radar el palillo queda oculto a simple vista tras la bola y, por lo tanto, éste es indetectable para el radar. En la figura 3.61 se muestra la matriz construida para realizar las medidas.

Una vez ubicada la estructura en el plano del objetivo, se ha realizado un barrido completo de la misma. Para tener una buena resolución en las medidas, se ha ejecutado el barrido a baja velocidad. En la tabla 3.8 se muestran los parámetros utilizados en la captura.

Parámetro	Valor
Velocidad de rotación del espejo	750 rpm
Velocidad de barrido vertical	50 steps/s
Versión del motor de elevación	'C'
Frecuencia de muestreo del ADC	40 MHz
Tamaño de la captura	3.74 GB

Tabla 3.8: Medida de matriz de bolas de 5mm.

Dadas las limitaciones de capacidad de transferencia de la tarjeta ADC descritas en el apartado 5.1.2.1, la captura se ha tenido que realizar en 4 sesiones distintas; cada una de ellas corresponde a un barrido angular en elevación distinto, de forma que, la unión de las 4 capturas permite explorar toda la matriz.

En la figura 3.62 se muestra la matriz de bolas capturada. También se ha representado en color negro la ubicación del anillo focal. Para obtener esta imagen se ha realizado un

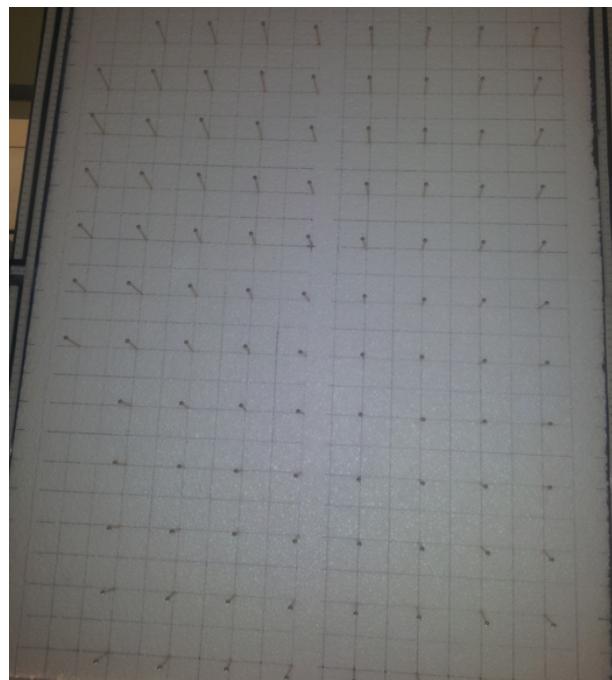


Figura 3.61: Sistema de coordenadas absolutas.

filtrado en distancia de las detecciones durante la etapa de procesado. Todo blanco que está a una distancia 3 cm más allá de la de las bolas ha sido eliminado de la escena; De tal modo, el panel de poliestireno queda oculto. La barra de color expresa en dBm la escala de potencia recibida.

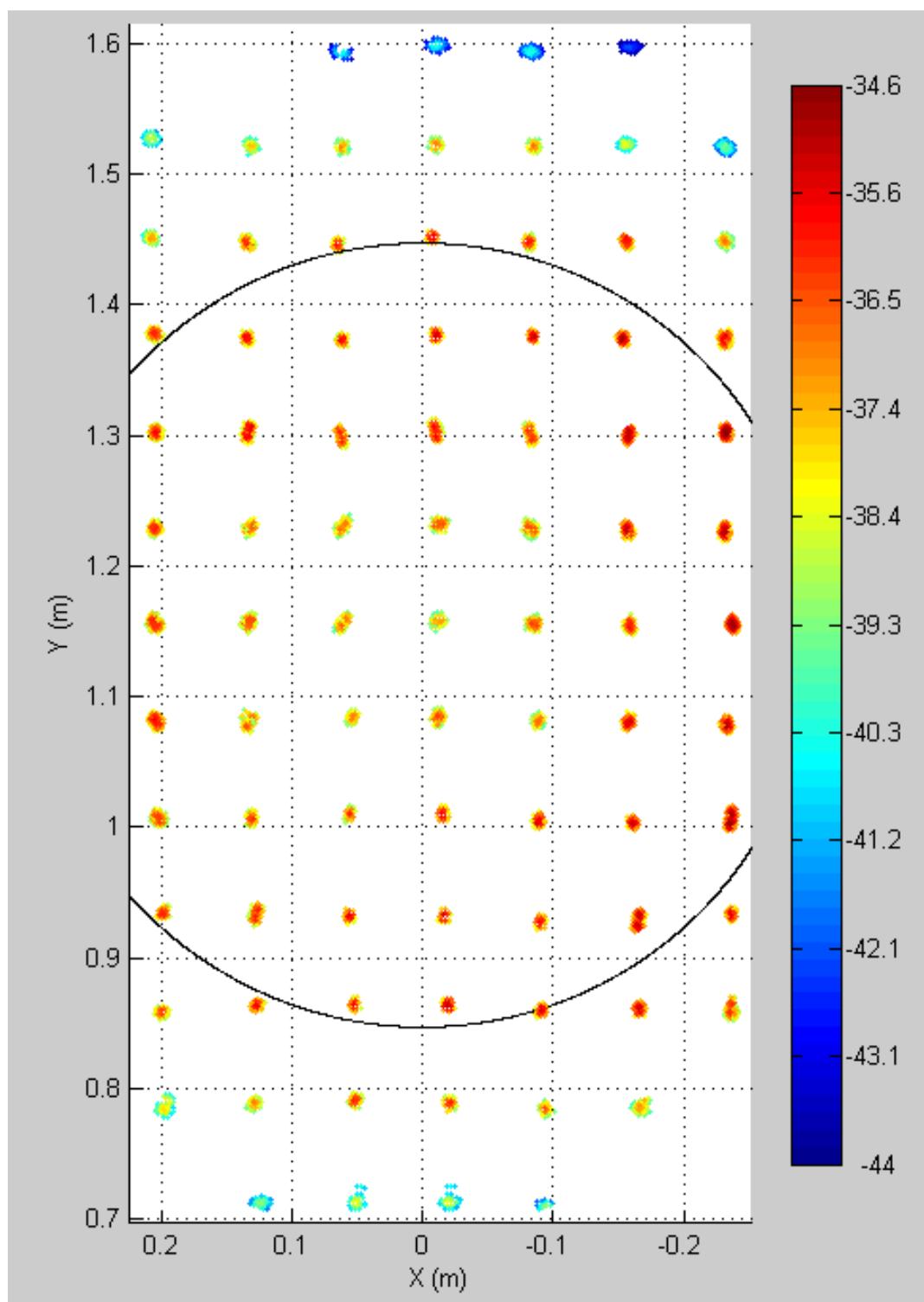


Figura 3.62: Matriz de bolas capturada.

En la figura 3.63 se muestra un detalle de la bola central de la figura 3.62.

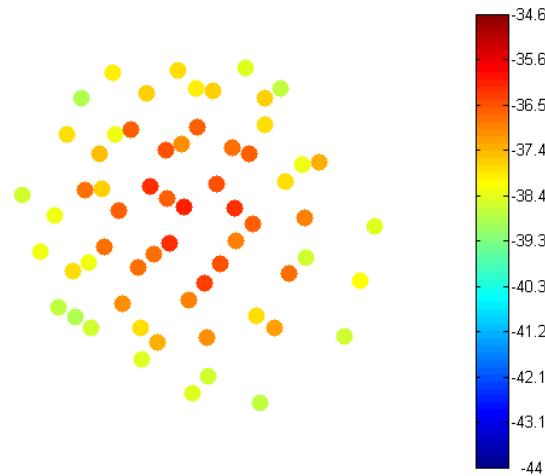


Figura 3.63: Detalle de la bola central de la matriz.

Se ha realizado posteriormente una interpolación de la figura anterior. La imagen obtenida se muestra en la figura 3.64.

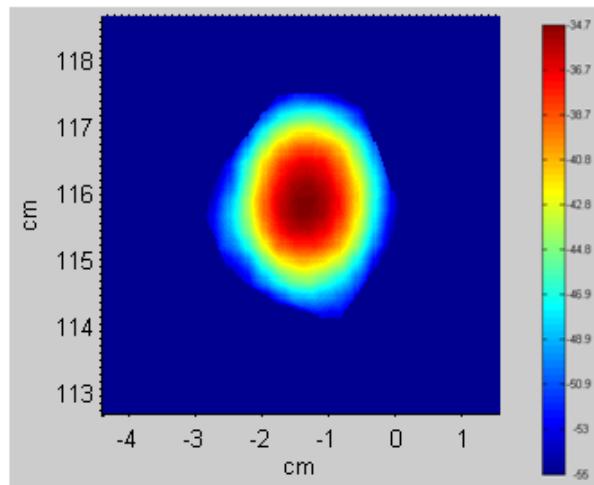


Figura 3.64: Detalle de la bola central de la matriz.

Respecto del punto de la esfera donde el nivel de potencia es máximo, se ha realizado un corte horizontal y otro vertical para analizar la forma del spot según se muestra en la figura 3.65.

A partir de dichos perfiles de potencia se determinan las dimensiones del spot siguiendo el criterio de corte a -3 dB. En el caso de la bola central, las dimensiones del spot son de 11 mm de ancho por 14 mm de alto.

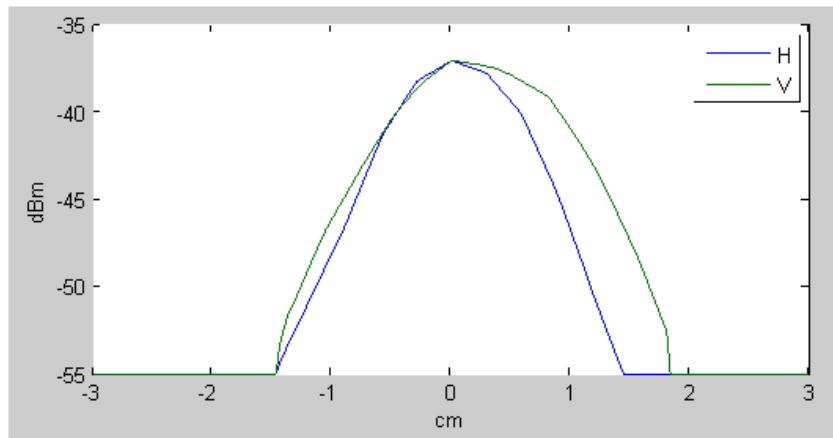


Figura 3.65: Perfiles de potencia vertical y horizontal del spot.

A continuación se muestra el resultado de las medidas de aquellas bolas que, por su posición o singularidades, se han considerado relevantes. La bola a la que se refiere cada medida se indica mediante un punto amarillo sobre la matriz de bolas. La potencia indicada indica el máximo nivel de potencia interpolado en la esfera. La potencia relativa se refiere respecto de la máxima potencia detectada en toda la matriz.

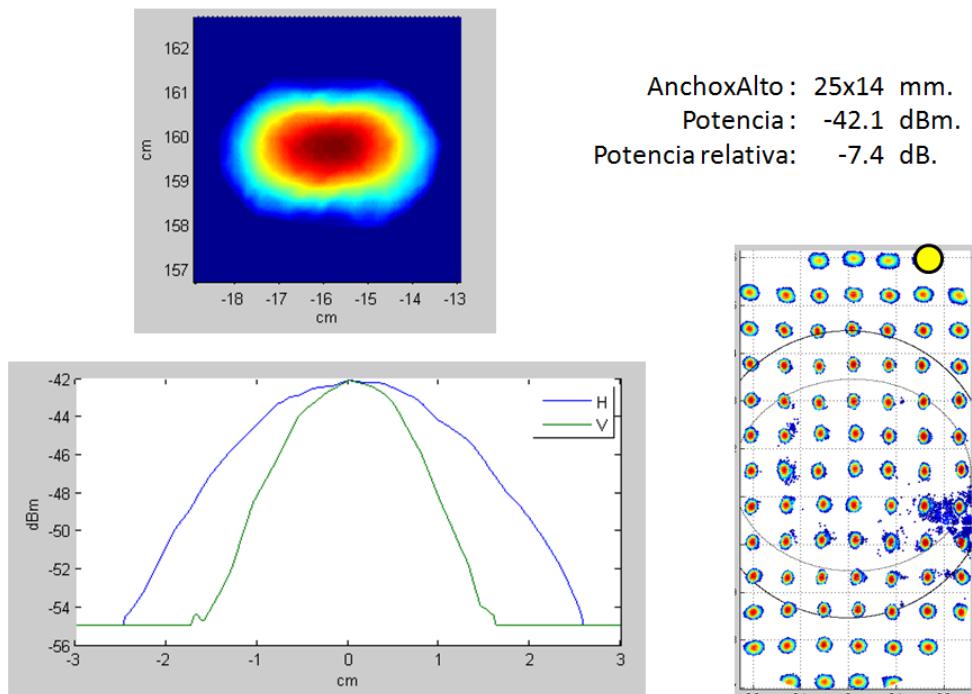


Figura 3.66: Medida de la bola de 5mm de la fila 1, columna 6.

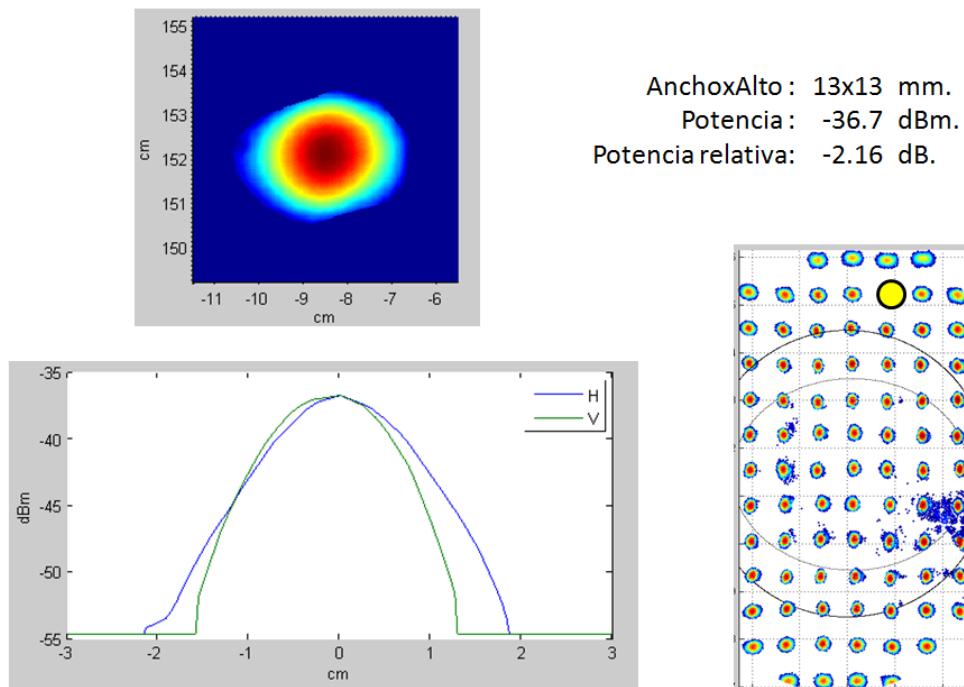


Figura 3.67: Medida de la bola de 5mm de la fila 2, columna 5.

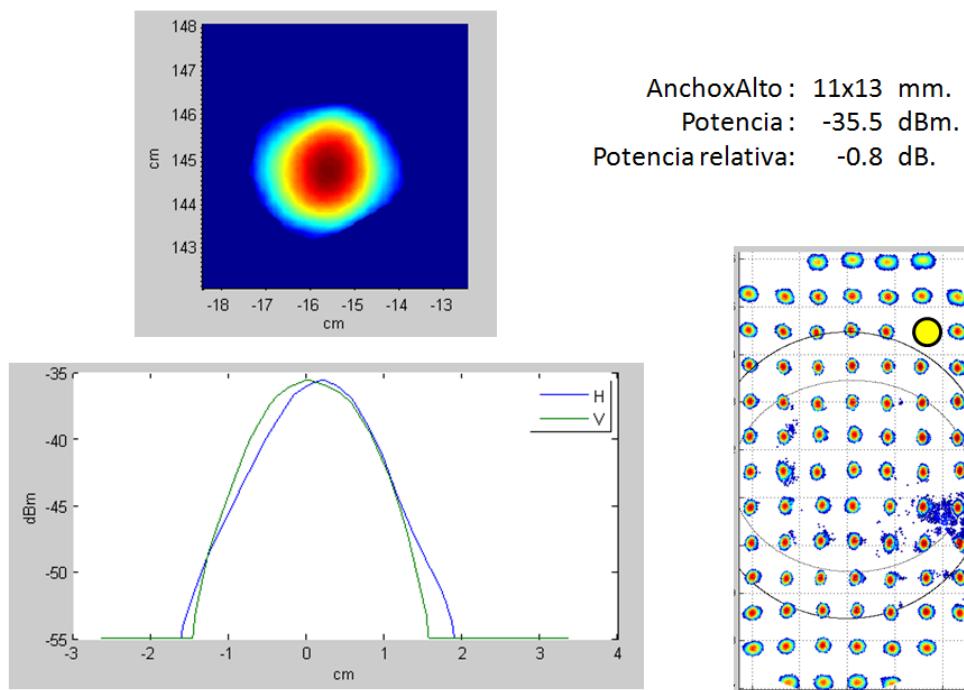


Figura 3.68: Medida de la bola de 5mm de la fila 3, columna 6.

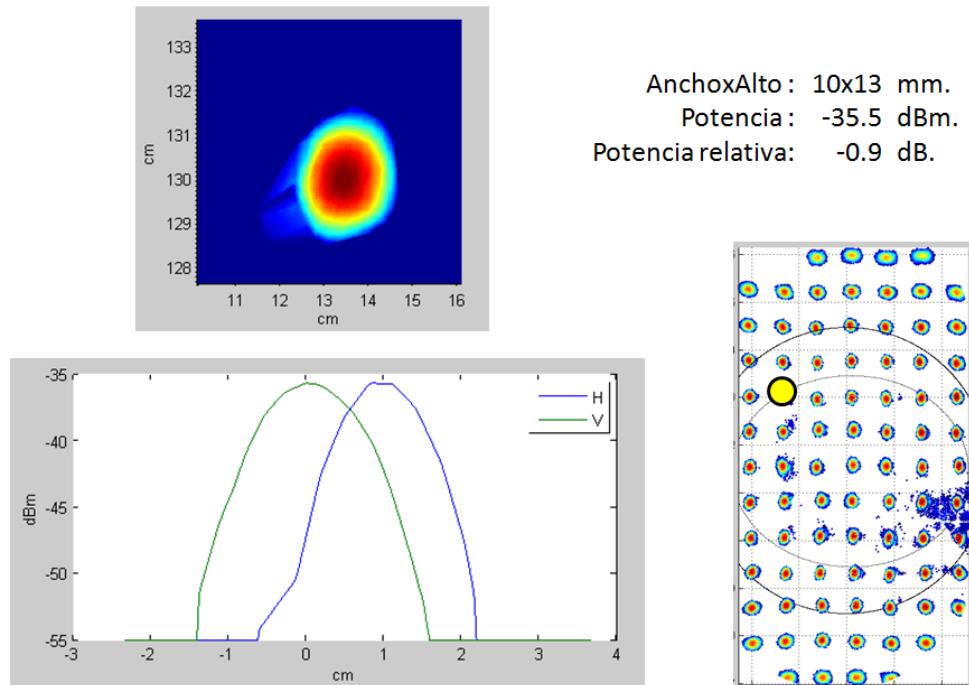


Figura 3.69: Medida de la bola de 5mm de la fila 5, columna 2.

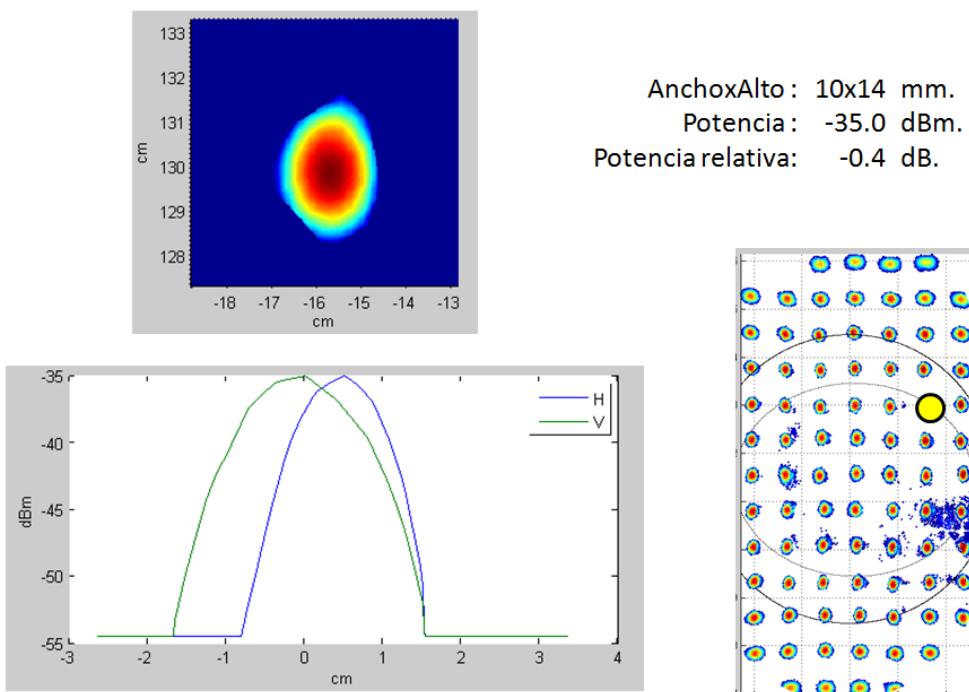


Figura 3.70: Medida de la bola de 5mm de la fila 5, columna 6.

3. Sistema de antenas.

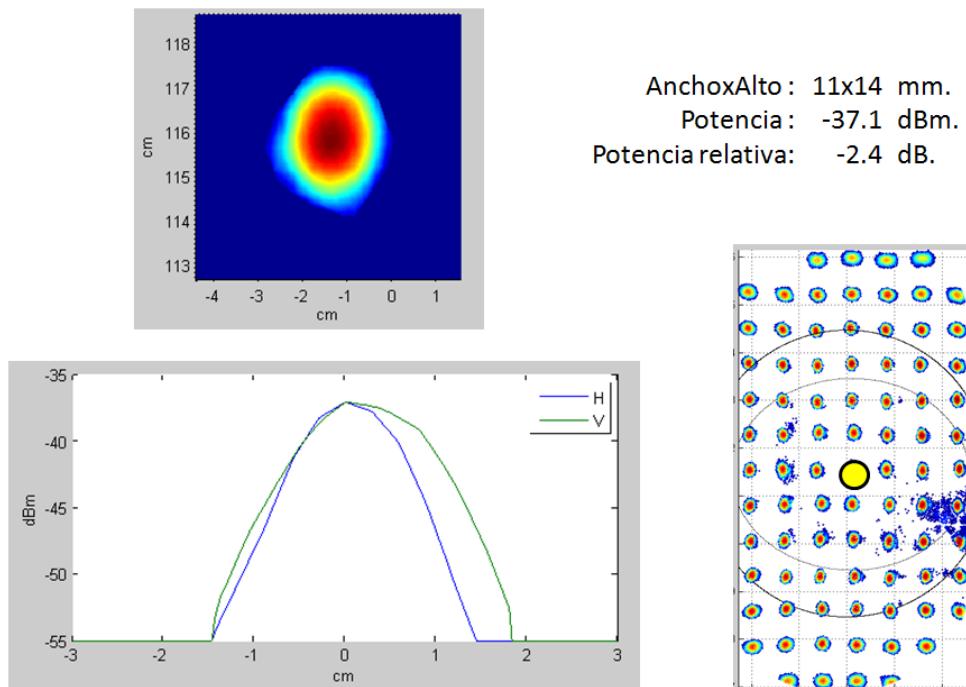


Figura 3.71: Medida de la bola de 5mm de la fila 7, columna 4.

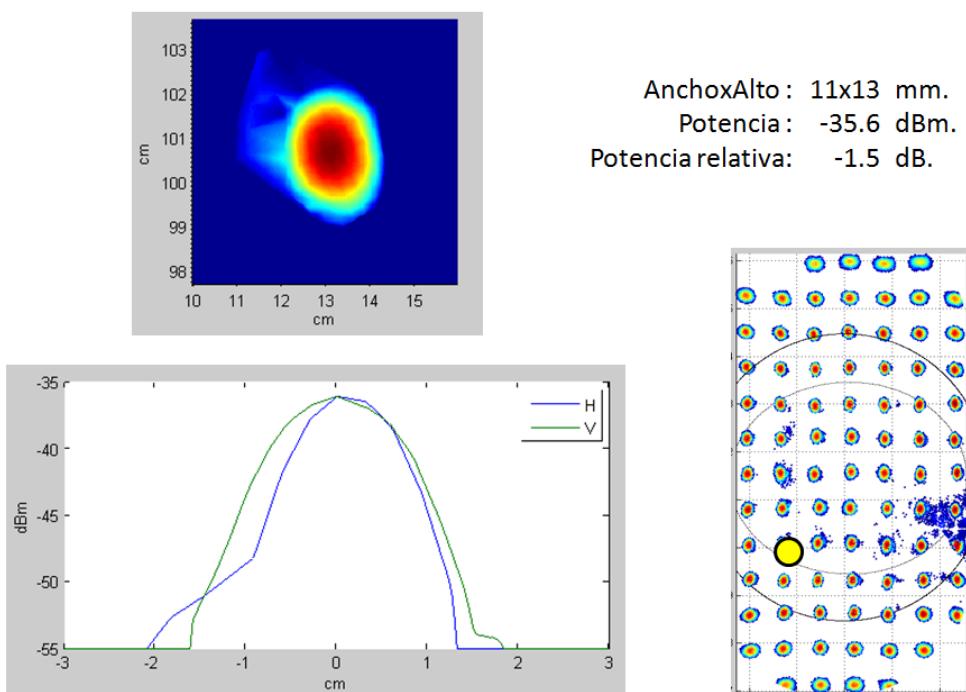


Figura 3.72: Medida de la bola de 5mm de la fila 9, columna 2.

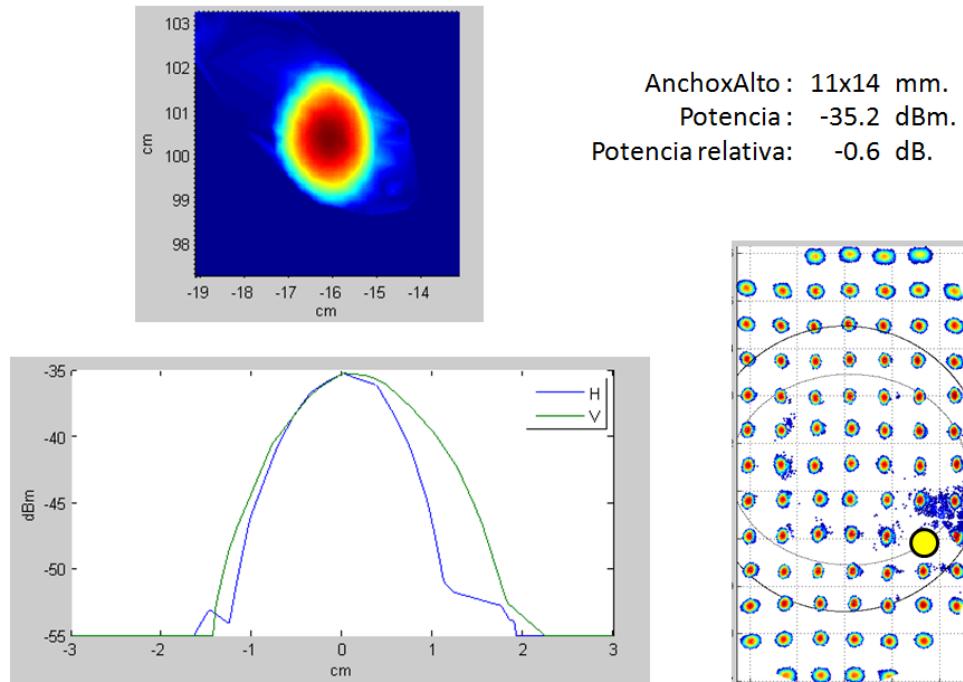


Figura 3.73: Medida de la bola de 5mm de la fila 9, columna 6.

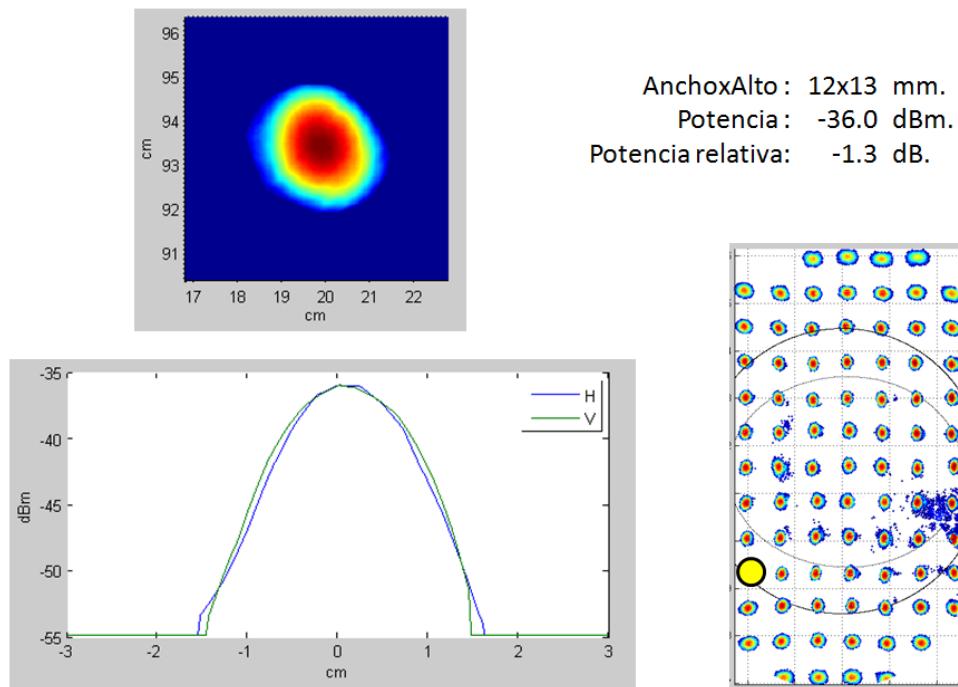


Figura 3.74: Medida de la bola de 5mm de la fila 10, columna 1.

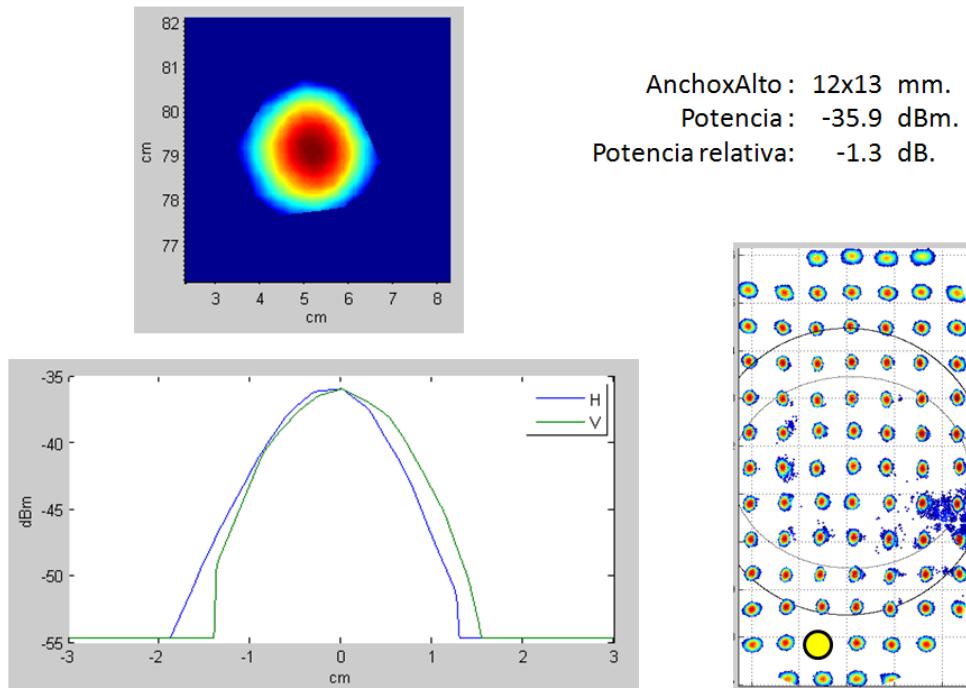


Figura 3.75: Medida de la bola de 5mm de la fila 12, columna 3.

Las figuras 3.76 y 3.77 muestran la variación de las dimensiones del spot. Los ejes coordenados indican el número de fila y columna de cada bola. La barra de color en ambas figuras está escalada en centímetros.

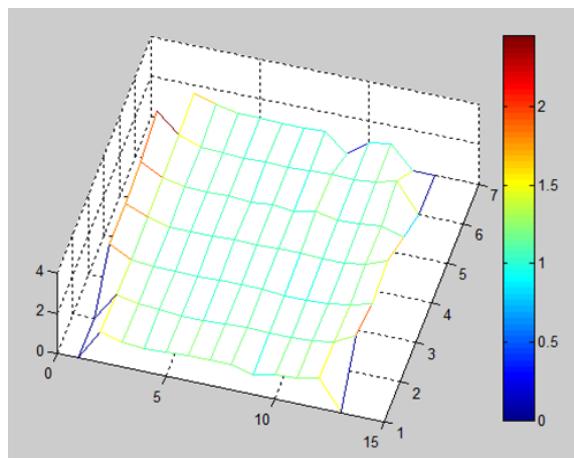


Figura 3.76: Anchura del spot.

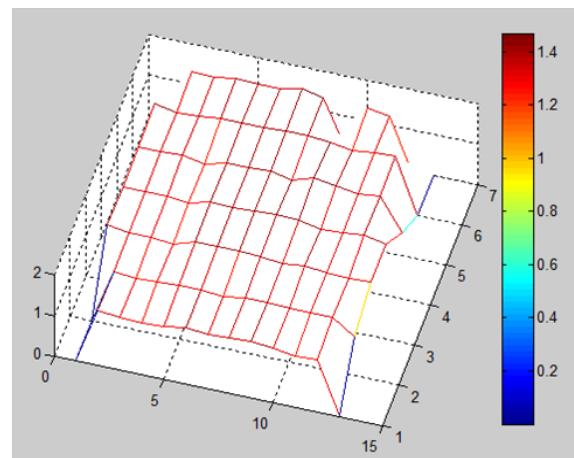


Figura 3.77: Altura del spot.

Se han realizado medidas de la misma matriz de bolas a diferentes distancias para comprobar la capacidad de enfoque de la antena fuera del plano focal. Se han procesado las medidas únicamente con la primera de las 4 partes que componen la matriz completa. En la figura 3.78 se muestra la zona superior de la matriz a una distancia de 8 metros

donde se puede observar como la diferencia de potencia entre los spots superiores y los centrales (excluyendo la primera fila) tienen un nivel de potencia similar.

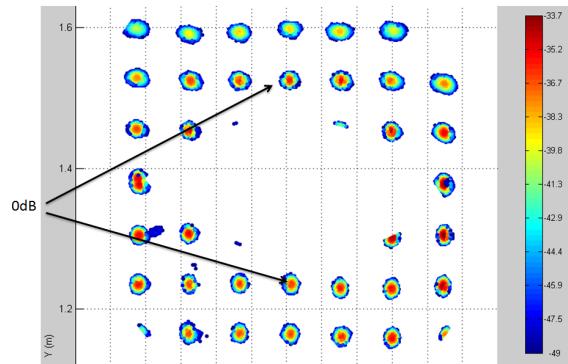


Figura 3.78: Matriz a 8 m del foco primario de la antena.

En la figura 3.79 se muestra la misma zona superior de la matriz alejada 22 cm del plano focal. Se puede apreciar como los spots centrales pierden nivel mientras que los superiores lo ganan. En la figura 3.80 la matriz se acerca 15 cm respecto del plano focal hasta los 7.85 m. Se aprecia cómo se produce el efecto contrario. Sin embargo, ni en el enfoque alejado ni en éste último, la forma del spot varía significativamente.

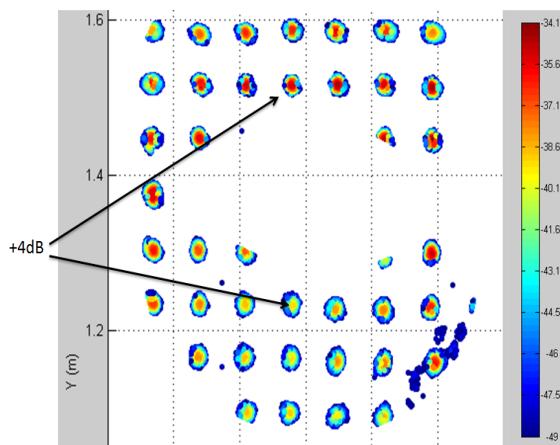


Figura 3.79: Matriz a 8.22 m del foco primario de la antena.

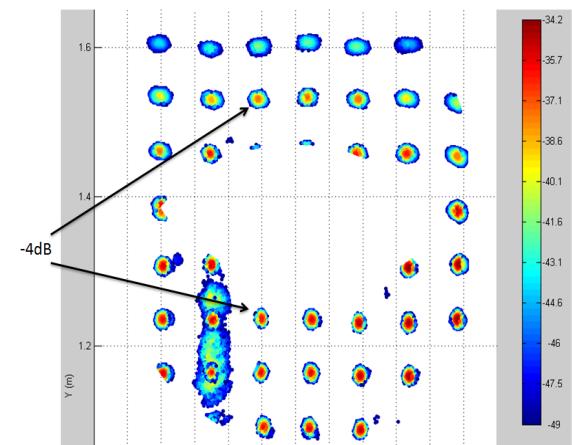


Figura 3.80: Matriz a 7.85 m del foco primario de la antena.

Capítulo 4

Sistema de generación de señal de radiofrecuencia.

4.1. Introducción.

En un radar la resolución en distancia viene determinada por el ancho de banda transmitido. Cuanto mayor es el ancho de banda transmitido mayor resolución en distancia se obtiene. Por ello, el sistema radar necesita de un dispositivo que genere una señal con un elevado ancho de banda.

La estrategia seguida para obtener dicho ancho de banda se basa en un DDS (Direct Digital Synthesizer). Se dispone de dos alternativas para generar la señal: DDS de Euvís y DDS de Analog Devices. Ambos generan una señal que posteriormente será elevada en frecuencia hasta los 300 GHz.

4.2. DDS Euvís.

El DDS del fabricante Euvís y modelo DSM303B se muestra en la figura 4.1. El DDS tiene un acumulador de fase de 32 bits y dispone de un conversor digital analógico de 11 bits. Es capaz de sintetizar frecuencias que se rigen por la fórmula $\frac{f_{RELOJ}}{A}$ donde A es el valor de un registro binario de 32 bits. Incorpora una memoria RAM de $511 \cdot 2^{10}$ (511K) registros de 32 bits y es capaz de sintetizar una única frecuencia o bien generar una señal chirp lineal o arbitraria. El rango de frecuencias que puede sintetizar varía entre DC y $\frac{f_{RELOJ}}{2}$ con una potencia comprendida entre -4 y 0 dBm.

El DDS es capaz de actualizar el tono que genera con una frecuencia $f_{actualización} = \frac{f_{reloj}}{8}$. La señal de reloj de entrada se introduce mediante conector SMA. Debe tener una frecuencia superior a 1.0 GHz e inferior a 3.0 GHz (2.5 GHz para el caso de generación chirp) y una potencia limitada entre +3 y +18 dBm.



Figura 4.1: *Direct Digital Synthesizer DSM303B.*

Dispone de un conector SMA por el que emite un pulso denominado señal *marker* que se asocia con el inicio de cada rampa. También incorpora un conector SMA por el que puede recibir una señal de disparo o *trigger* la cual se activa por un flanco de bajada.

El *DDS* cuando genera señales chirp soporta 2 modos de trabajo: continuo y a ráfagas. En el modo continuo el *DDS* genera la forma de onda programada de forma periódica. El inicio de este modo puede venir determinado por un comando software o por un *trigger* externo.

En el modo a ráfagas la generación de señal se inicia cuando se detecta un *trigger* externo. Un contador programable de 8 bits determina el número de veces que se ha de repetir la forma de onda programada. La señal *trigger* es ignorada hasta que el *DDS* no termina de ejecutar la ráfaga. El periodo mínimo de trigger soportado y, por tanto, de la ráfaga es de $1 \mu\text{s}$.

En el modo continuo la forma de onda con el menor periodo permitido tiene 32 registros. En cada registro se indica al *DDS* la frecuencia a sintetizar durante los 8 siguientes ciclos de reloj. En el modo a ráfagas el mínimo periodo es de 256 ns. Este valor es un parámetro fijo que no depende de la frecuencia del reloj de entrada.

El *DDS* implementa una máquina de estados interna. Dependiendo del estado en el que se encuentra dicha máquina, el tiempo que transcurre desde que se recibe el *trigger* externo hasta que se activa la señal ***marker*** y, por tanto, se comienza a generar la rampa programada puede variar. Por esta razón, este modo de trabajo no es aconsejado si los tiempos de guarda entre rampas no garantizan que la máquina de estados retorna siempre al mismo estado.

Presenta un ruido de fase a 10 KHz de la portadora de -145 dBc/Hz y dispone de una memoria interna de 520,000 registros de 32 bits cada uno donde se almacena la forma de onda a sintetizar. Es capaz de almacenar en su memoria hasta 128 for-

mas de onda diferentes. Las especificaciones técnicas del equipo están disponibles en <http://www.euvitis.com>.

El fabricante facilita una *GUI* (figura 4.2) para la programación del mismo aunque también se puede hacer uso de la API desarrollada para los lenguajes basados en CLR (Common Language Runtime) tales como Visual Basic, Visual J#, Visual C# y Visual C++. La comunicación entre el equipo programador y el *DDS* se realiza mediante un interfaz USB. Actualmente se emplea esta *GUI* para configurar el DDS. Está previsto implementar la API dentro de la aplicación del sistema en un futuro próximo.

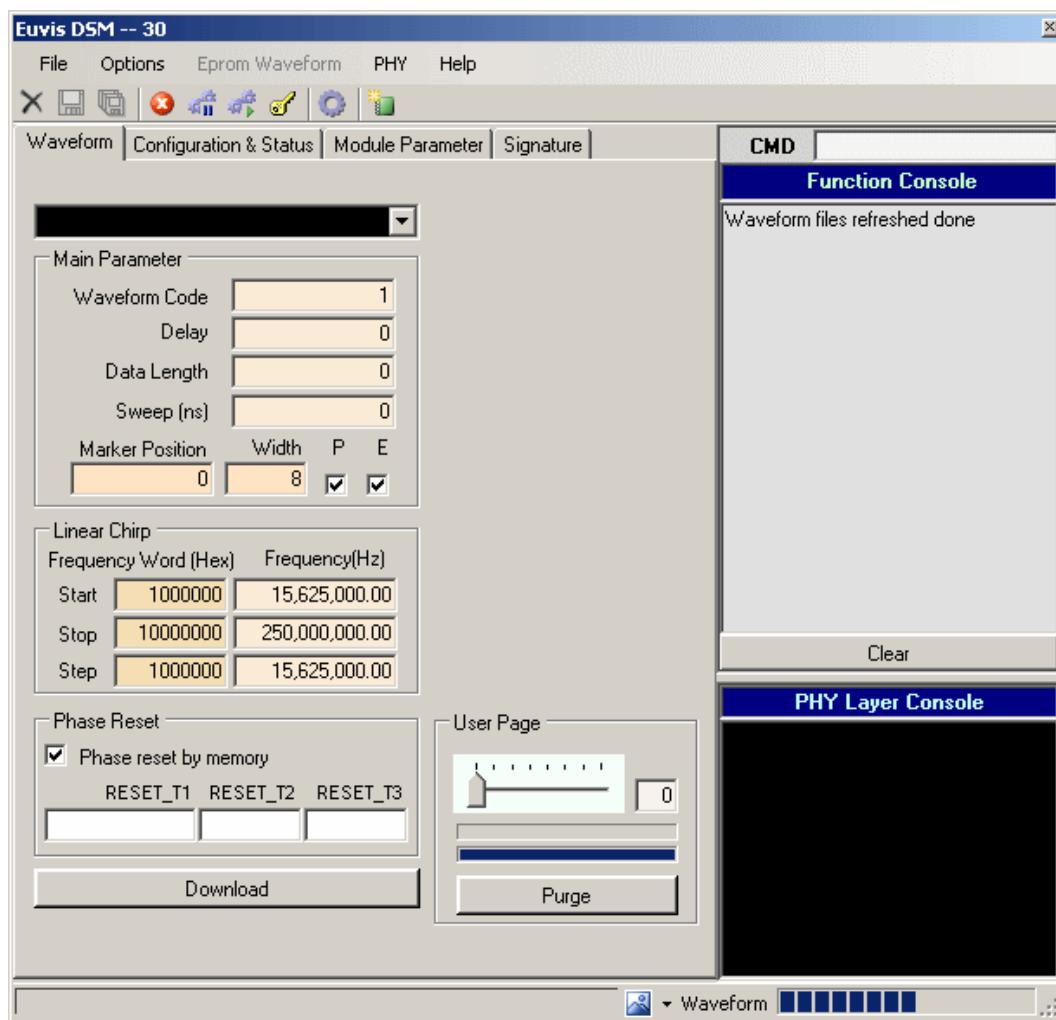


Figura 4.2: *Direct Digital Synthesizer DSM303B*.

El DDS implementado en el radar utiliza como fuente de reloj la señal generada por el sintetizador de 2.4 GHz. Con esta frecuencia de reloj se pueden generar rampas con un periodo mínimo de $32 \text{ registros} \cdot \frac{1}{f_{actualización}} = 32 \cdot \frac{1}{2.4\text{GHz}} \cdot 8 = 106\text{ns}$ y con un periodo máximo de $511 \cdot 2^{10} \text{ (511K) registros} \cdot \frac{1}{f_{actualización}} = 511K \cdot \frac{1}{2.4\text{GHz}} \cdot 8 = 1.74\text{ms}$.

Actualmente, este DDS trabaja en el radar generando una señal *chirp* lineal en modo continuo y disparado por software con un **periodo de rampa** de $150 \mu\text{s}$ en el cual se transmite un ancho de banda de 500 MHz. La señal generada se ubica en el margen de frecuencias comprendido entre los 500 MHz y 1 GHz.

EL *DDS* incorpora 3 fuentes conmutadas del fabricante Texas Instruments y modelos PTN78060A PTH08080W Y PTH12050W similares a la mostrada en la figura 4.3 que generan respectivamente tensiones de -5V, +5V y +1.8V a partir de la única tensión de alimentación de 12 V que acepta la tarjeta.



Figura 4.3: Fuente conmutada PTH12050W.

Estas fuentes introducen espurios a la salida de la señal generada por el *DDS* cuyas componentes espectrales coinciden con las frecuencias de conmutación de las mismas (550 KHz, 300 KHz y 320 KHz respectivamente). Dichas componentes se confunden con blancos en la etapa de procesado. Por ello, las fuentes mostradas en la figura 4.4 han sido retiradas de la tarjeta y sustituidas por fuentes lineales externas.

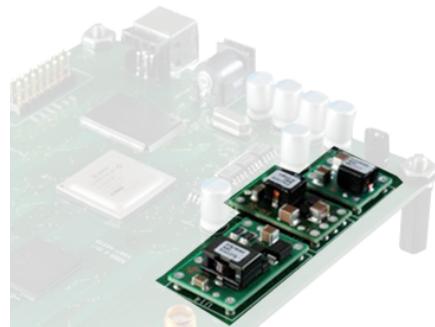


Figura 4.4: Fuentes retiradas de la tarjeta.

Capítulo 5

Sistema de control y sincronismo.

Como se ha descrito en el apartado 1.4, para obtener una imagen radar es necesario que determinados elementos del sistema trabajen de forma sincronizada. Dichas funciones son realizadas por el sistema de control y sincronismo.

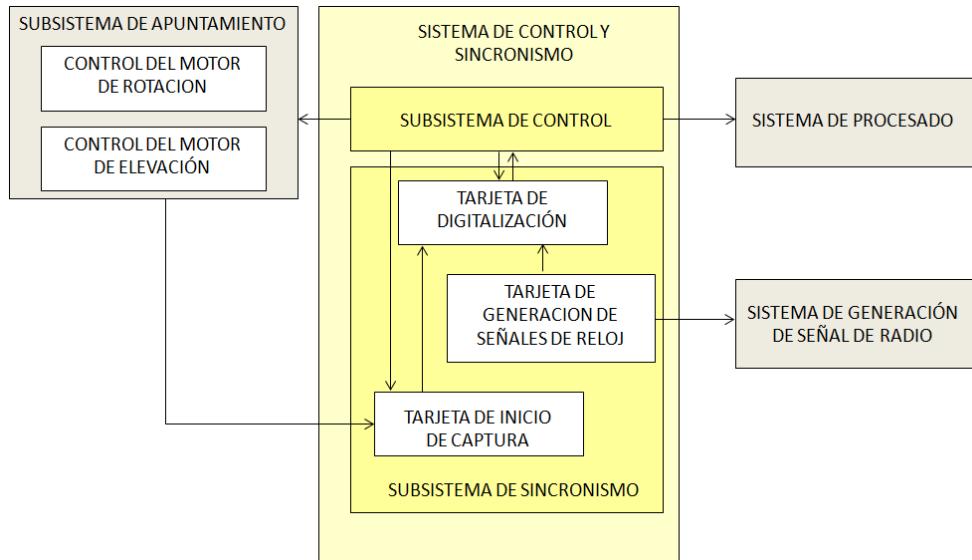


Figura 5.1: Diagrama del sistema de control y sincronismo.

El sistema de control y sincronismo se estructura, según se muestra en la figura 5.1, en dos subsistemas: el subsistema de control y el de sincronismo. El subsistema de control se encarga de proporcionar una HMI (Human Machine Interface) mediante una aplicación gráfica que permite al usuario configurar la forma de trabajo del escáner.

El subsistema de sincronismo, que se configura por el subsistema de control, implementa los mecanismos necesarios para lograr una determinada correspondencia entre la señal de radiofrecuencia generada, la dirección de apuntamiento del sistema de antenas y las muestras capturadas por el *ADC* (véase 5.1.2).

5.1. Subsistema de sincronismo.

El radar obtiene información de la distancia de los posibles blancos asociados a un *spot* transmitiendo una señal *chirp* lineal. La señal transmitida se mezcla con la recibida obteniéndose una señal denominada **señal de batido** que es digitalizada por el ADC. Mediante la aplicación del algoritmo *FFT* (*Fast Fourier Transform*) a la muestras asociadas a una rampa chirp de la señal de batido, se obtiene una señal en el dominio de la frecuencia denominada **perfil de distancias** por cada rampa chirp donde cada pico de energía detectado se corresponde con la reflectividad de cada blanco. Cuanto mayor es la frecuencia en la cual se detecta el pico de energía mayor es la distancia del blanco detectado. En las figuras 5.2 y 5.3 se muestra un ejemplo de señal de batido del radar en cuestión en el dominio del tiempo y en el de la frecuencia. La señal obtenida corresponde a una dirección de apuntamiento donde el sistema de antenas enfoca a una lámina de poliespan de 2 cm de espesor ubicada a 8 metros de distancia.

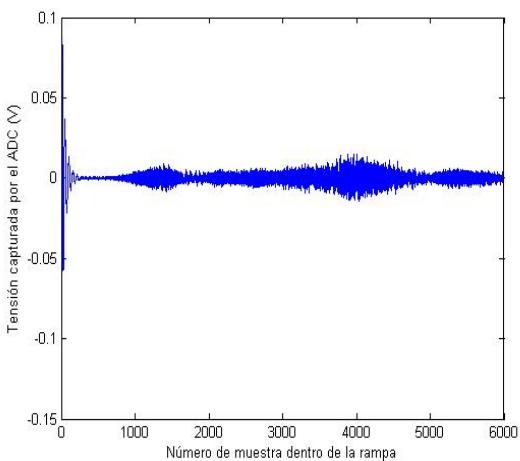


Figura 5.2: Señal de batido capturada.

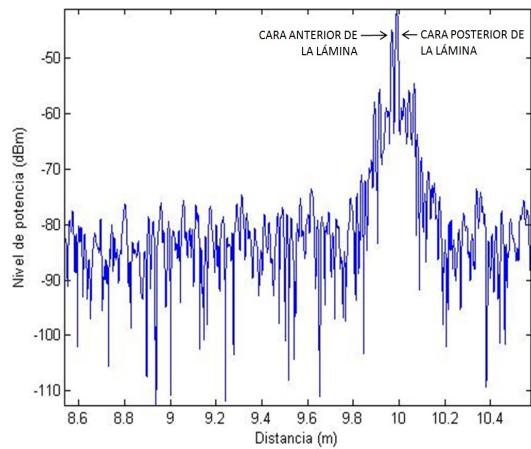


Figura 5.3: FFT de la señal de batido.

Para lograr obtener una imagen se necesita obtener un perfil de distancias de cada spot del plano del objetivo. Ello se consigue logrando una correspondencia unívoca entre direcciones de apuntamiento de la antena, inicios de rampa de cada señal *chirp* y muestras capturadas por el ADC. Una posible estrategia para obtener la imagen puede ser la de posicionar el espejo de exploración en cada dirección de apuntamiento requerida y posteriormente generar una señal de disparo tanto al DDS como al ADC, para que, el primero transmita una rampa chirp y el segundo inicie la captura de la señal de batido. Esta estrategia tiene el inconveniente del retardo variable de disparo del DDS descrito en el apartado 4.2. A este inconveniente hay que añadirle la limitación que tiene el triplicador de frecuencia que eleva la frecuencia de 50 a 150 GHz de la cadena del oscilador local, el cual no acepta una variación brusca de la potencia de la señal de entrada y por ello dispone a su entrada de un atenuador de ajuste manual según se muestra en la figura 5.4. Además, los motores en este modo de funcionamiento invierten

demasiado tiempo en posicionar el espejo. Por todo ello, se ha optado por otra opción que hace trabajar al DDS generando rampas chirp sucesivas y mover adecuadamente ambos motores de forma que se logre un barrido completo de la escena según se detalla en la sección 3.1.9. De esta forma se minimiza el tiempo de exploración.

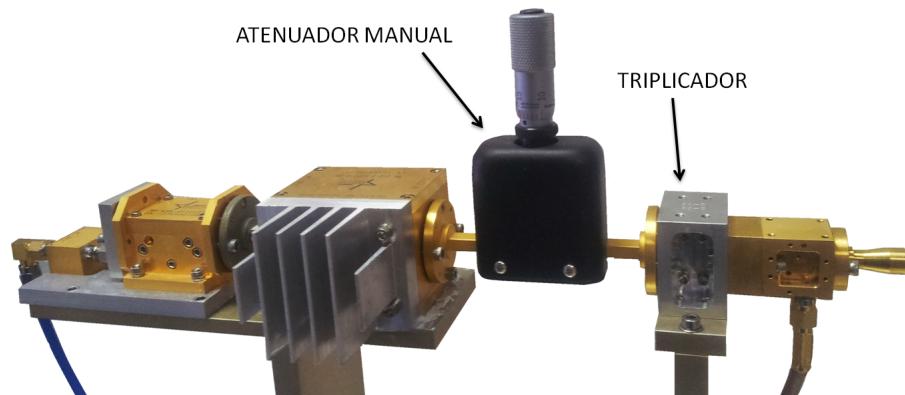


Figura 5.4: Atenuador de ajuste manual a la entrada del triplicador.

En este modo de trabajo es necesario conocer la orientación del espejo en cada inicio de rampa de la señal chirp. Para ello, se dispone de dos posibilidades: capturar la información facilitada por los encoders de ambos motores en cada instante de inicio de una rampa, o bien, asociar la muestra inicial capturada por el ADC con una orientación concreta del espejo de exploración y, suponiendo una velocidad de giro constante de ambos motores, estimar la orientación del espejo en cada inicio de rampa. La primera opción implica una mayor captura de información. Además, si se inicia la captura con el espejo de exploración en reposo las primeras muestras estarán más separadas que las últimas ya que el motor de rotación tarda un tiempo hasta alcanzar la velocidad demandada por el controlador; Por ello, se ha recurrido a la segunda opción. En ella, los motores del subsistema de apuntamiento se mueven a gran velocidad para lograr que el tiempo de muestreo de la escena sea reducido.

El motor de rotación debe girar a una velocidad en torno a las 8,000 rpm para lograr imágenes en un tiempo inferior al segundo. Si el motor parte de una situación de reposo en el instante de inicio de la captura el controlador del motor no podrá imponer tanta aceleración al motor y no se podrá realizar la aproximación de velocidad constante supuesta. Por esta razón, en el instante de inicio de la captura, el motor deberá rotar previamente a la velocidad de trabajo. De esta manera, el proceso de captura consiste en activar la generación periódica de la señal chirp, disminuir la atenuación del atenuador manual gradualmente hasta su posición de trabajo, posicionar el motor de elevación en uno de los extremos del ángulo γ de barrido vertical e iniciar el movimiento del motor de rotación. Cuando el motor alcanza la velocidad de trabajo, un circuito de detección de posición inicial del espejo debe generar una señal trigger al controlador del motor de elevación para que éste comience a realizar el barrido vertical de la escena que se va a capturar. La misma señal trigger debe disparar la captura de la tarjeta digitalizadora.

De tales funciones se encarga la tarjeta de inicio de captura descrita en el apartado 5.1.3.

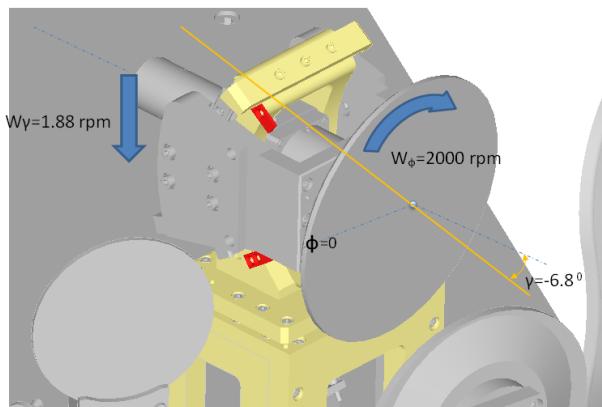


Figura 5.5: Movimiento del espejo de exploración.

La señal de reloj del ADC es la referencia temporal que sirve al subsistema de sincronismo para estimar la dirección de apuntamiento del sistema de antenas. Es necesario tener una única referencia temporal que garantice que cada rampa chirp digitalizada contenga el mismo número de muestras capturadas con el ADC. Para ello, el subsistema de sincronismo incorpora un generador de señales de reloj que se describe en el apartado 5.1.1.

El inicio de la generación de la señal chirp no es un proceso síncrono, es decir, se realiza cuando el operador lo estima oportuno. Por ello, cuando la tarjeta de inicio de captura detecta la posición inicial del espejo del exploración y genera la correspondiente señal trigger al ADC y al controlador del motor de elevación, el DDS puede estar sintetizando cualquier punto de su rampa de frecuencia. Para conocer el instante en el cual se comienza a generar una rampa, el ADC captura por un segundo canal la señal marker del DDS descrita en el capítulo 4 y que genera un pulso en cada inicio de rampa. En la figura 5.6 se muestra un esquema del proceso completo.

Durante las pruebas se han detectado problemas de ruido causados por la acción de los motores que se inducen en la electrónica del sistema de control y sincronismo. Para subsanar estos problemas se ha tenido que separar la alimentación de los motores de la del resto de los sistemas; implementando para ello una fuente conmutada y asilando las masas de una y otra fuente. También se han instalado cables apantallados en la mayor parte de los cables de los motores y se han instalado condensadores a la entrada de la alimentación de los controladores de los motores. A pesar de todas estas medidas que atenúan el ruido, se han tenido que realizar modificaciones en el diseño del circuito de disparo de inicio de captura para que éste último sea inmune a dicho ruido.

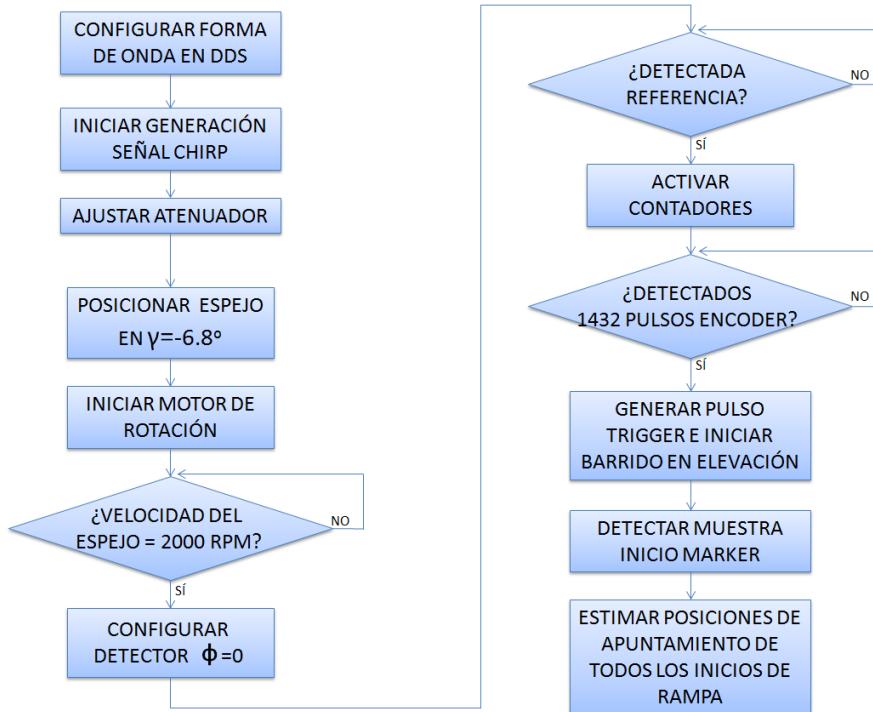


Figura 5.6: Diagrama de flujo del sistema de sincronismo.

5.1.1. Generador de señales de reloj.

El escáner necesita de distintas señales de reloj para poder operar. Se necesitan relojes de 40MHz, de 500 MHz y de 2.4 GHz (Véase la sección 5.1.2 y el capítulo 4 respectivamente). La señal de 500 MHz se ha implementado para poder sustituir el DDS por otro descrito en el apéndice C. Todas estas frecuencias se obtienen a partir de una única señal de 10 MHz denominada **base de tiempos del sistema** según se muestra en la figura 5.7.

Todos los dispositivos necesarios para generar tales señales de reloj se integran, al igual que la *base de tiempos del sistema*, en la caja de generación de señales (véase el apartado 5.1.1.6).

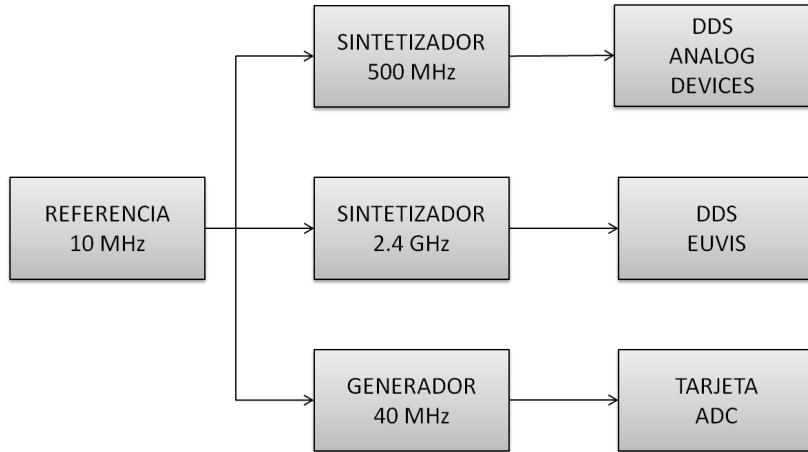


Figura 5.7: Reloj del sistema.

5.1.1.1. Señal de reloj de 10 MHz.

La base de tiempos del sistema viene determinada por una referencia de precisión de 10 MHz. Se genera a partir de un reloj de cuarzo calefactado de la marca Oscilloquartz y modelo OCXO8653 (figura 5.8) que se integra en la caja de generación de señales (véase 5.1.1.6).



Figura 5.8: Cristal de cuarzo de 10 MHz.

El cristal se polariza a través de un conector DB9 con una tensión de 24 V y requiere de una potencia 8W. Una vez que el elemento calefactor ha alcanzado la temperatura de trabajo el consumo de potencia disminuye por debajo de los 4W.

El cristal genera una señal sinusoidal de +6 dBm de potencia con una estabilidad a corto plazo (entre 1 y 10 segundos) inferior a $2 \cdot 10^{-12}$. A largo plazo la estabilidad es inferior a $1.5 \cdot 10^{-10}$ por cada año de funcionamiento. El nivel típico de armónicos generado es de -40 dBc. Los espurios cuyas frecuencias son inferiores a 1 MHz tienen unos niveles máximos de -80 dBc. En la figura 5.9 se muestra el ruido de fase de la señal generada.

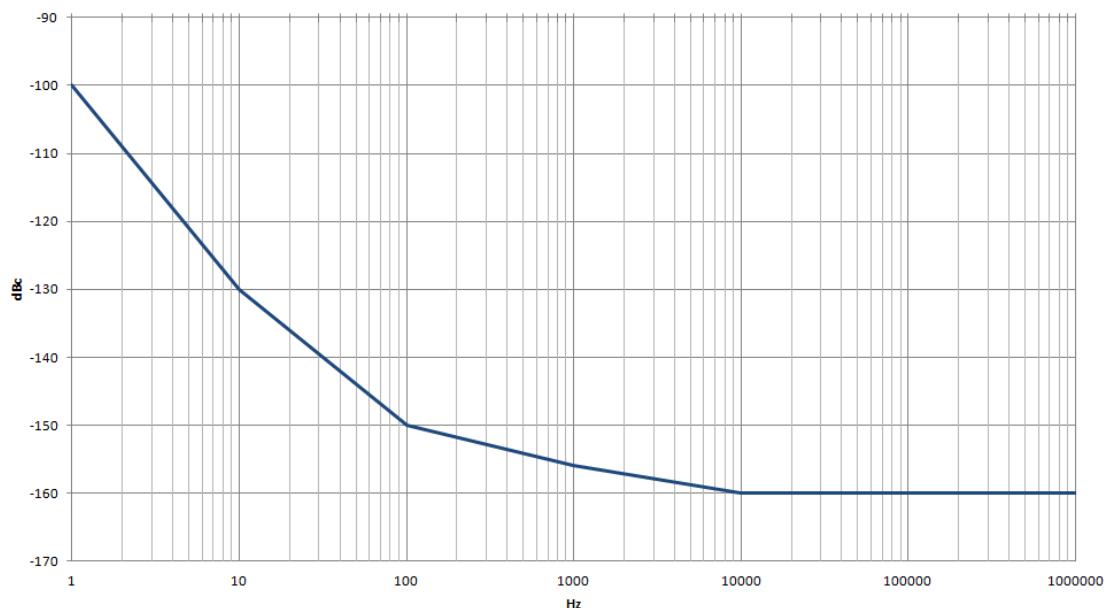


Figura 5.9: Ruido de fase.

Según se muestra en la parte superior izquierda de la figura 5.10, la señal de salida del reloj del sistema se atenúa en un divisor resistivo con una estructura en π (R501,R502 y R503). Posteriormente la señal es amplificada (U501) e introducida en un circuito híbrido (U502) de tipo *power splitter* de la marca Minicircuits y modelo AD4PS. A la salida de dicho circuito se obtienen 4 réplicas de la señal de entrada que sirven de señal de referencia a los relojes de 2.4 GHz, 500 MHz y 40 MHz.

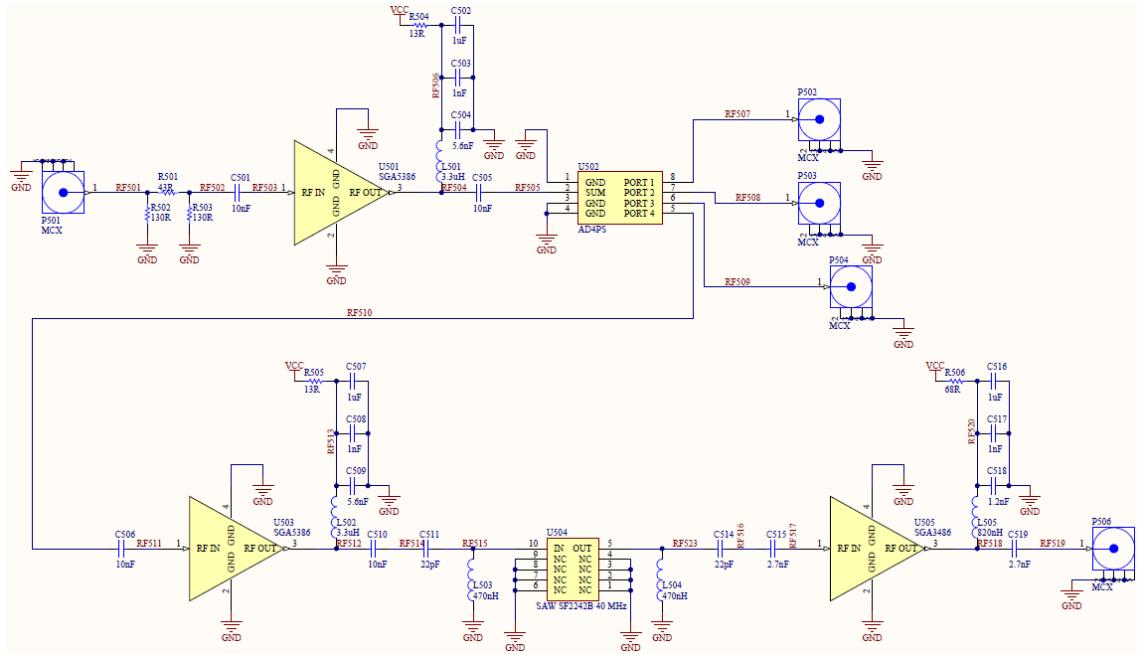


Figura 5.10: Generación de señales de referencia.

5.1.1.2. Señal de reloj de 2.4 GHz.

El dispositivo encargado de generar la señal de 2.4 GHz es un sintetizador de la marca Synergy y modelo LFSW110250-50. Dicho sintetizador es programable mediante un puerto *SPI* (*Serial Peripheral Interface*) integrado en el mismo. El sintetizador es capaz de generar frecuencias entre 1.1 y 2.5 GHz con una resolución de 500 KHz a partir de una señal de 10 MHz con un reducido ruido de fase mostrado en la figura 5.11. La potencia de salida es de 5 dBm. Tiene una supresión de espurios de 75dB y una atenuación en las frecuencias armónicas de 15 dB.

Para lograr que éste genere 2.4 GHz a partir de una señal de 10 MHz se ha de enviar la siguiente cadena de bytes al puerto *SPI*: *0x4D30303030393630*. La programación almacenada no es persistente y por lo tanto es necesario reenviar dicha cadena de bytes cada vez que el sintetizador se polariza.

El fabricante facilita una placa de evaluación, según se muestra en la figura 5.12, dotada de un conector DB25 por el que recibe alimentación y se conectan todas sus entradas y salidas a excepción de la señal de referencia y la de salida de RF, ambas terminadas mediante un conector SMA. Incorpora también un interfaz que convierte el puerto I2C en determinadas líneas de un puerto paralelo a fin de que el sintetizador pueda ser programado con un equipo que disponga de un puerto *LPT*.

El fabricante también facilita una aplicación dotada de *GUI* (véase la figura 5.13) que permite una programación sencilla del dispositivo.

Sin embargo, el puerto paralelo es hoy en día una interfaz obsoleta. Por esta razón,

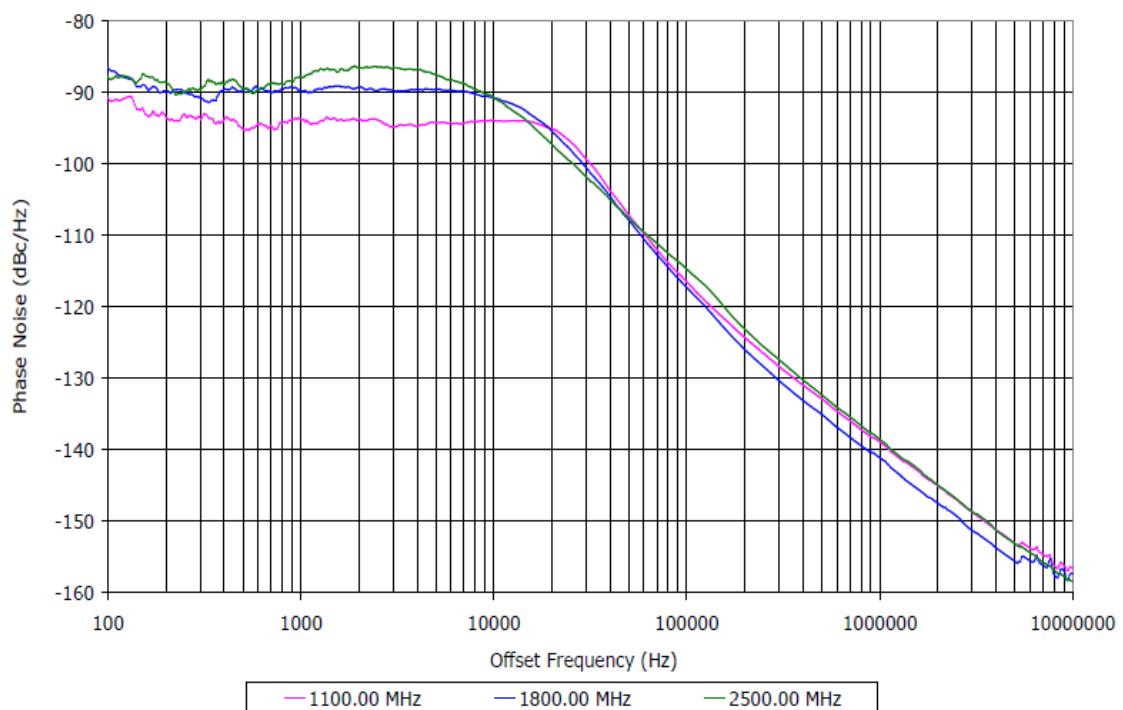


Figura 5.11: Ruido de fase.



Figura 5.12: Placa de evaluación para sintetizadores Synergy.

se ha diseñado una PCB, denominada **programador de la tarjeta de evaluación**, dotada de un conector DB25 que se acopla a la tarjeta de evaluación anteriormente indicada según se muestra en la figura 5.14.

La *PCB* incorpora un microcontrolador del fabricante MICROCHIP y modelo PIC-16F84A que envía la cadena de bytes de configuración al sintetizador. Para más detalle sobre dicha implementación véase el apéndice B.2. Actualmente el radar no utiliza la tarjeta de evaluación ya que se ha desarrollado una nueva tarjeta (véase 5.1.1.5) que integra directamente el sintetizador.

5.1.1.3. Señal de reloj de 500 MHz.

La señal de reloj de 500 MHz se ha generado mediante otro sintetizador de la marca Synergy y modelo LFSW35105-100. Este sintetizador utiliza también como señal de

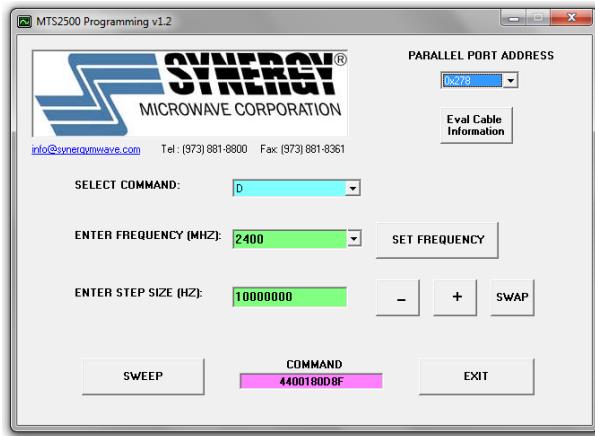


Figura 5.13: Software para la programación de sintetizadores Synergy.

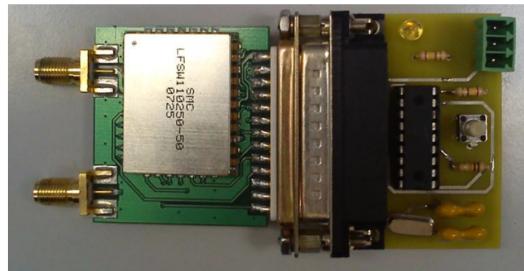


Figura 5.14: Programador de la tarjeta de evaluación.

reloj de entrada una señal de 10 MHz y es capaz de generar frecuencias en el intervalo [350,1050] MHz con una resolución de paso de 1000 KHz. El sintetizador garantiza una supresión de espurios de 85 dB y de armónicos de 20 dB. La potencia de salida de la señal sintetizada es de 10 dBm. El sintetizador presenta un reducido ruido de fase mostrado en la figura 5.15.

Al igual que el sintetizador de 2.4 GHz, el dispositivo debe de ser configurado mediante el envío de una cadena ASCII (American Standard Code for Information Interchange) a través de su interfaz *SPI*. El dispositivo se integra dentro de la tarjeta de síntesis (véase la sección 5.1.1.5).

5.1.1.4. Señal de reloj de 40 MHz.

La señal de reloj de 40 MHz se obtiene utilizando como referencia la base de tiempos del sistema. Dicha señal se amplifica mediante un amplificador MMIC (Monolithic Microwave Integrated Circuits) de silicio y germanio (SiGe) de la marca RF Micro Devices y modelo SGA-5386 (véase U503 en la figura 5.10)

El amplificador, que tiene una ganancia de 17 dB a la frecuencia de 10 MHz, trabaja en modo saturado comportándose como un dispositivo no lineal. Por tal motivo, la señal

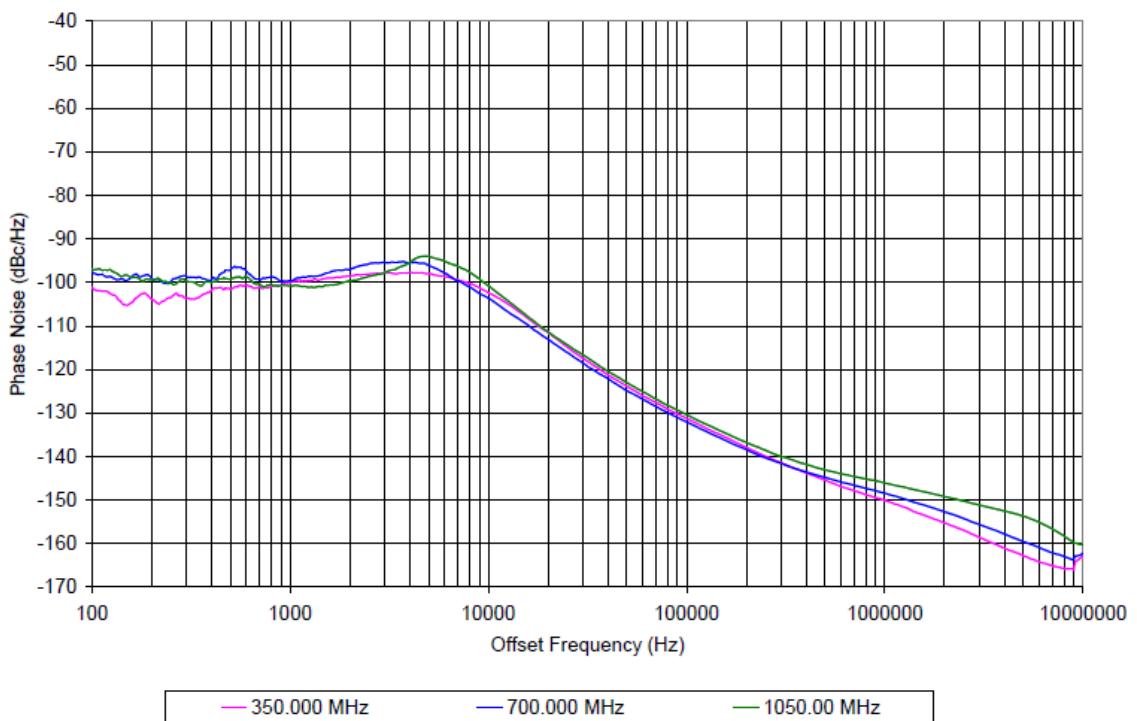


Figura 5.15: Ruido de fase.

de la salida del dispositivo contiene en sus componentes espectrales armónicos de la señal de entrada.

La señal de salida es filtrada (véase U504 en la figura 5.10) mediante un filtro paso banda de tipo SAW (Surface Acoustic Wave) de la marca RF Monolithics y modelo SF2242B. El filtro, cuya frecuencia central es de 40 MHz, tiene un ancho de banda (utilizando el criterio de frecuencia de corte a 3 dB) de 5 MHz. En la figura 5.16 se muestra la respuesta del filtro en frecuencia.

El diseño del circuito de amplificación y el posterior filtro no forma parte del alcance del proyecto. La implementación de dichos esquemas en un circuito impreso sí forma parte de las labores realizadas en este proyecto.

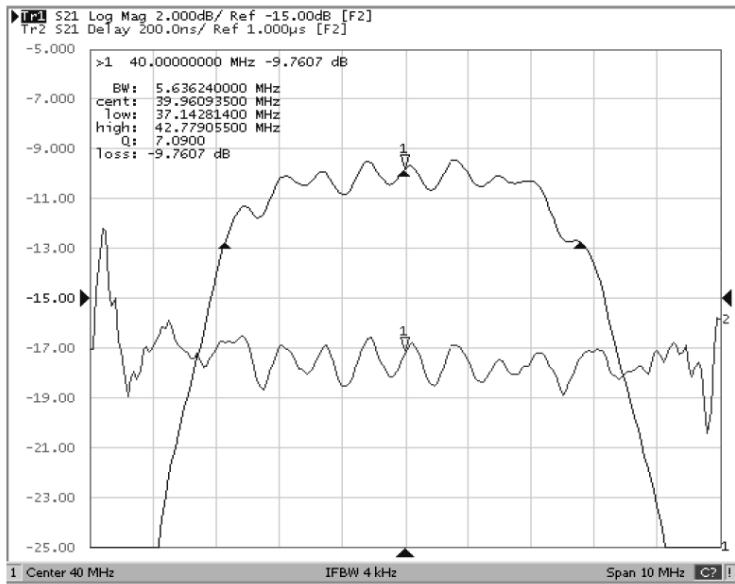


Figura 5.16: Respuesta en frecuencia del filtro SAW.

5.1.1.5. Tarjeta de generación de señales de reloj.

La tarjeta (figura 5.17) dispone de los siguientes elementos (entre paréntesis se muestra el nombre del componente especificado en los esquemas de la tarjeta y que están disponibles en el apéndice B):

- Un conector de polarización (P302) por el que recibe una tensión de 5V.
- Un sintetizador de 500 MHz (U201) y elementos de polarización del mismo.
- Un conector MCX (P201) por el que se recibe la referencia del *sintetizador 500*.
- Un filtro paso bajo (L203 etc...) a la salida del *sintetizador 500*.
- Un circuito híbrido (U202) que duplica la señal de 500 MHz filtrada.
- Un conector MCX (P202 y P203) por cada una de las dos salidas del circuito híbrido a 500 MHz.
- Un sintetizador de 2.4 MHz (U101) y elementos de polarización del mismo.
- Un conector SMA (P101) por el que se recibe la referencia del *sintetizador 2.4*.
- Un conector IDC de 16 pines (P301) para la interconexión con el MOD5270 (véase B.1).
- Una fuente de alimentación de 3.3V (U301) para polarizar el MOD5270.

- Un conector MCX (P501) por el que se recibe una señal de 10 MHz.
- Un amplificador para una señal (U501) de 10MHz de 17 dB de ganancia junto con sus respectivos elementos de polarización.
- Un circuito híbrido (U502) que cuadriplica la señal amplificada de 10 MHz.
- Tres conectores MCX (P502, P503 y P504) que reciben 3 de las 4 salidas del cuadruplicador.
- Otro amplificador igual al anterior (U503) que se conecta en la salida disponible del cuadruplicador y que trabaja en la zona no lineal para generar, entre otros, un armónico de 4º orden.
- Un filtro *SAW* de 40 MHz (U504) conectado a la salida del amplificador no lineal.
- Un amplificador a 40MHz (U505) de 21 dB de ganancia que recibe la señal de salida del filtro *SAW*.
- Un conector MCX (P506) que transmite la señal de 40MHz filtrada.
- Un conector de salida de 5V (P303) en el que se conecta el ventilador de la caja de generación de señales (véase 5.1.1.6).
- Un conector MCX que transmite una señal digital de *trigger* generada por el MOD5270.

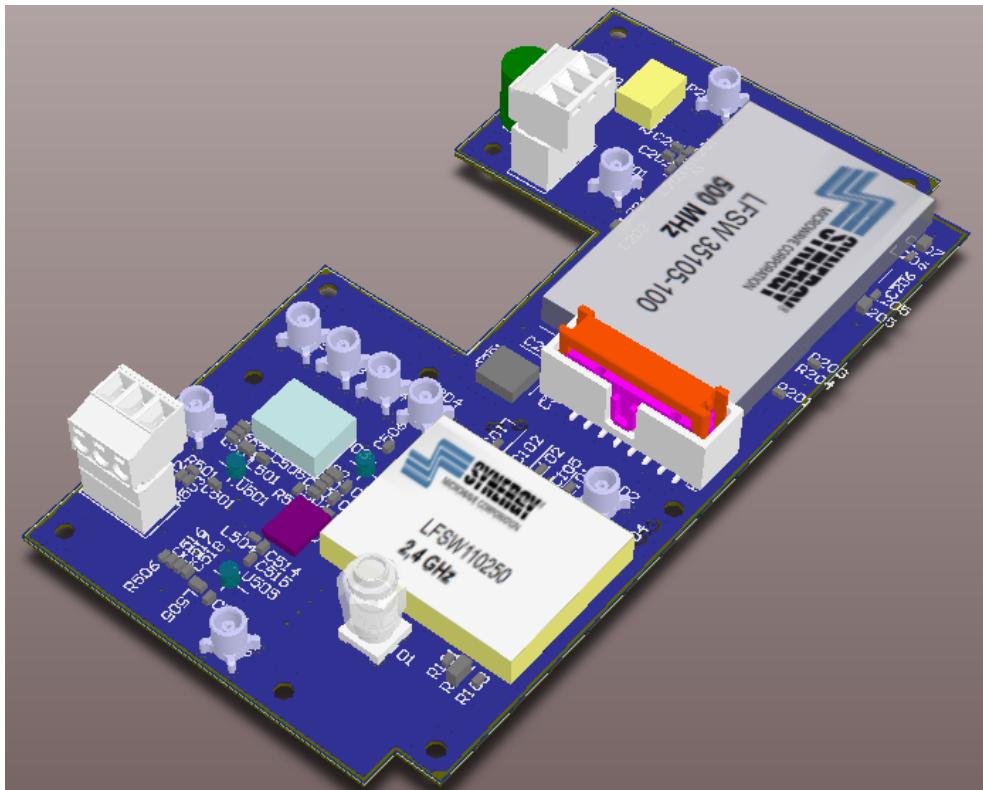


Figura 5.17: Tarjeta de síntesis generación de señales de reloj.

La *PCB* se ha fabricado sobre un sustrato Cuclad de 0.254 milímetros de espesor cuya permitividad es de 2.17 F/m . Ambas caras del dieléctrico están recubiertas por una capa de cobre de $17 \mu\text{m}$. Para darle consistencia al sustrato éste se ha soldado mediante indio a una lámina de latón de 1 mm, mecanizada según se muestra en la figura 5.18.

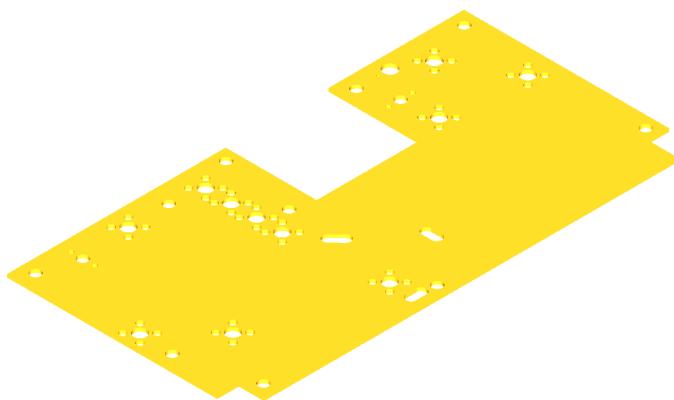


Figura 5.18: Mecanización de la lámina de latón.

En este nuevo diseño los sintetizadores se programan mediante un módulo MOD5270 que se ubica en la caja de generación de señales. Dicho módulo incorpora un microcon-

trolador dotado de un controlador de bus *SPI* (véase B.1).

A pesar de que el microcontrolador del MOD5270 dispone de un controlador de bus *SPI*, los sintetizadores han tenido que ser programados utilizando directamente líneas del bus de datos del microcontrolador incorporado en el módulo en una estrategia similar a la empleada en el *programador de la tarjeta de evaluación*.

La razón para no usar el controlador *SPI* integrado se debe a que el sistema operativo que se ejecuta sobre el micro interrumpe en ocasiones la transferencia del mensaje *SPI*. Dicha posible pausa esta contemplada en el estándar *SPI* pero los sintetizadores seleccionados no la toleran.

El MOD5270 se acopla a una PCB fabricada en un sustrato *FR4* según se muestra en la figura 5.19. Tiene un pulsador de *reset*, un conector IDC de 16 pines para la interconexión con la tarjeta de generación de señales de reloj y un transceptor de bus que intercala las líneas de datos utilizadas en la programación de los sintetizadores y que sirve para proteger de posibles daños al bus del microcontrolador. Los esquemas circuitales están disponibles en el apéndice B.4.

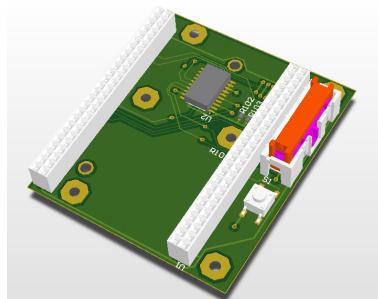


Figura 5.19: PCB para la inserción del módulo MOD5270.

El modulo MOD5270 se comporta de manera similar al programa implementado en el *programador de la tarjeta de evaluación*. Cuando la tarjeta de generación de señales de reloj es polarizada se encienden los sintetizadores y el microcontrolador. Medio segundo más tarde, el microcontrolador envía por los pines *DATA_IN* y *CLOCK_IN* (pines 5 y 11 de P301 respectivamente) el mensaje *SPI* con la configuración al sintetizador_500. Dichas señales llegan a los puertos *SPI* de ambos sintetizadores. Una tercera señal del bus y distinta para cada sintetizador *LATCH_ENABLE_500* y *LATCH_ENABLE_2.4* selecciona el sintetizador que se va a programar.

Tras enviar la cadena de configuración almacenada en la memoria no volátil del MOD5270 se activa la señal *LATCH_ENABLE_2.4* y se envía la cadena correspondiente al sintetizador_2.4.

El módulo MOD5270 está dotado de un puerto ethernet 802.3u. A través de él el micro y la aplicación del sistema (véase 5.2) pueden intercambiar mensajes *UDP* (*User Datagram Protocol*). De esta forma el subsistema de control puede solicitar al micro que reenvíe una determinada cadena de configuración a un sintetizador o bien indicarle a

éste que almacene una nueva cadena en memoria *flash* de forma que, cuando el sistema sea nuevamente polarizado, se transmita esta nueva cadena.

5.1.1.6. Caja de generación de señales.

La caja de generación de señales es de la marca RS y código de producto 517-4437. Es de aleación de aluminio ADC12 presofundido de 2 milímetros de espesor. En la figura 5.20 se muestra como la caja integra el reloj de 10 MHz (representado en color rojo), el *DDS* (representado mediante un rectángulo naranja), la tarjeta de síntesis (representada en color azul), el MOD5270 y su PCB (representados en color blanco y verde oscuro respectivamente), un filtro activo (representado en color verde claro), un ventilador (representado en color blanco) y conectores pasamuros.

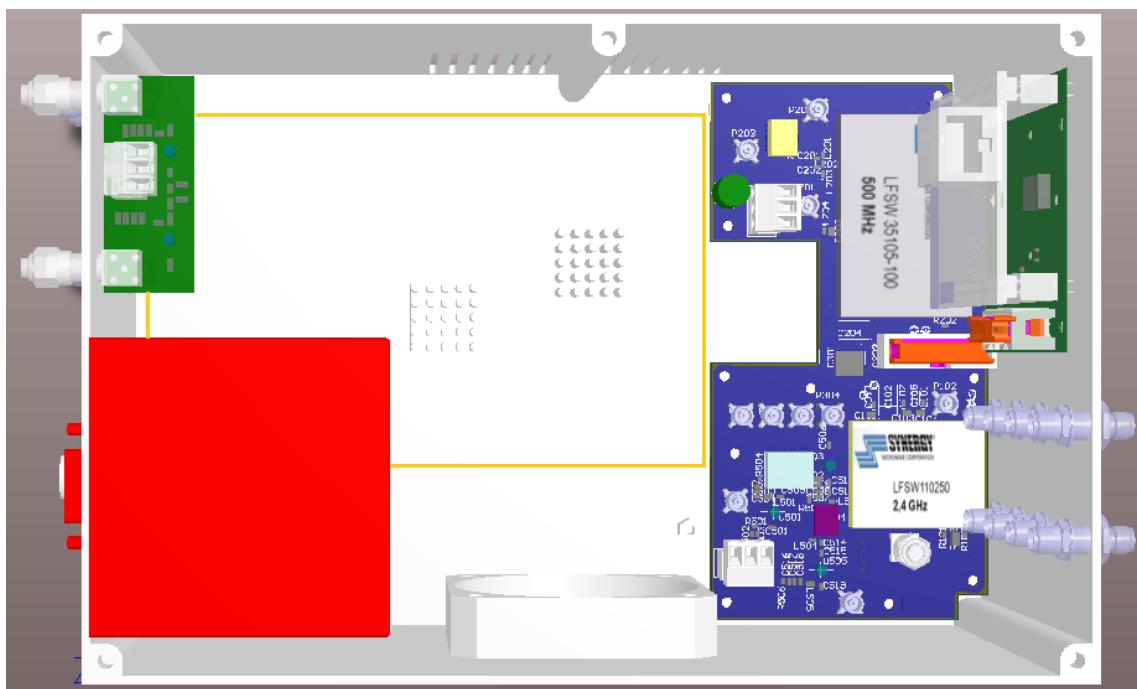


Figura 5.20: Caja de generación de señales I.

La caja ha sido mecanizada con unas ranuras para permitir la entrada de flujo de aire y con otra abertura para la expulsión de dicho flujo mediante la acción de un ventilador acoplado a la abertura. El ventilador se polariza con una tensión de 5 V que suministra la tarjeta de síntesis.

5.1.2. Tarjeta de digitalización.

El conversor analógico digital o **ADC**(*Analog to Digital Converter*) es el elemento encargado de digitalizar la señal de batido recibida para su posterior tratamiento.

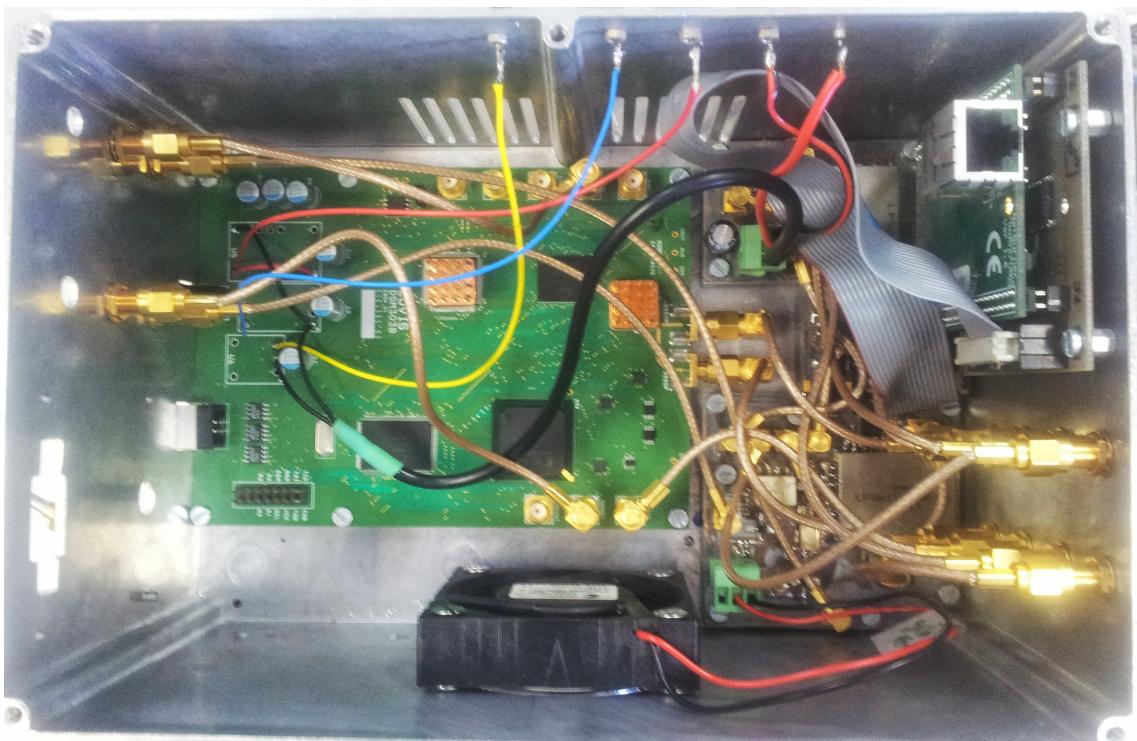


Figura 5.21: Caja de generación de señales II.

El conversor es de la marca ADLINK y modelo PCI-9846. Es capaz de muestrear simultáneamente hasta cuatro canales a razón de 40 Megamuestras por segundo y con una resolución de 16 bits.



Figura 5.22: Tarjeta ADC.

Incorpora un reloj interno de 40 MHz que utiliza para determinar los instantes de captura aunque el hardware del *ADC* también permite utilizar una señal externa como fuente de reloj. En tal caso, la señal de reloj se debe introducir por el conector SMB denominado *external clock*.

El otro conector SMB se denomina *external trigger* y permite sincronizar el inicio de

la captura (Véase D.1). También se puede determinar el inicio de la captura mediante *trigger* por software, o bien, recibir la señal de captura de otra tarjeta *ADC* a través del bus SSI según se muestra en la figura 5.23. Además de estas opciones, también se puede configurar un inicio de captura analógico. Un convertidor de 8 bits muestrea uno de los 4 canales. Cuando se supera un determinado valor se inicia la captura.. Los detalles de los modos de inicio de la captura se encuentran disponibles en el apéndice D.1. La tarjeta también permite realizar un diezmado de las muestras capturadas (véase el apéndice D.2).

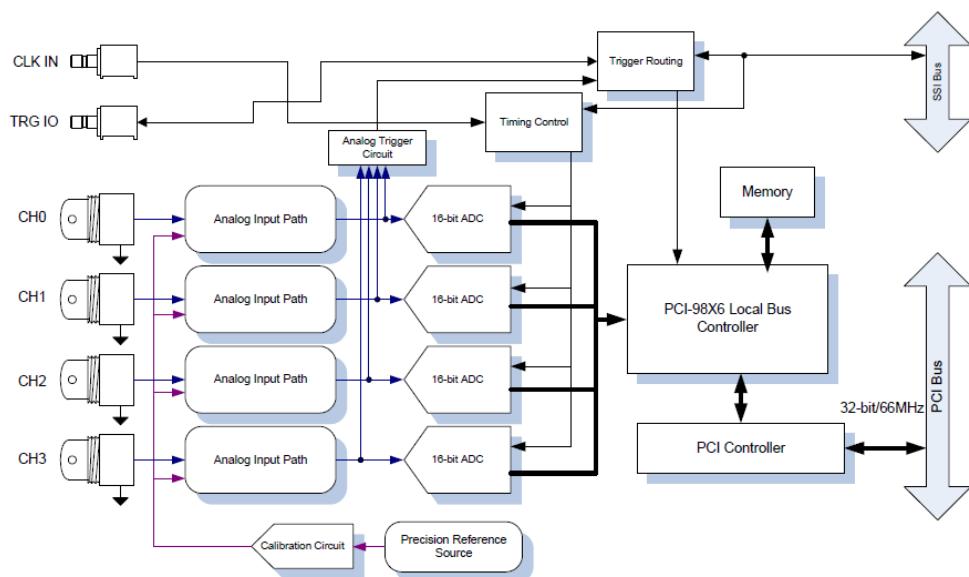


Figura 5.23: Diagrama funcional del ADC.

El *ADC* implementa una función de autocalibrado en la que utiliza una referencia interna de tensión de gran estabilidad térmica para la misma. En la tabla 5.1 se muestran las principales características técnicas del mismo.

Parámetro	Valor	Descripción
Impedancia	$50 \Omega / 1 M\Omega$	Ajustable individualmente por software
Rango de entrada	$\pm 0.2 \text{ V} / \pm 1 \text{ V}$	Ajustable individualmente por software
Diáfonia	<80 dB	En la banda de frecuencias [0,1] MHz.
Error offset	$\pm 0.2 \text{ mV}$	
Error ganancia	$\pm 0.1 \% / \pm 0.05 \%$	Respectivamente para escalas $\pm 0.2 \text{ V}$ y $\pm 1 \text{ V}$
Memoria interna	512 MB SDRAM	

Tabla 5.1: Especificaciones técnicas del ADC.

En la figura 5.24 se muestra el diagrama de bloques de uno de los circuitos de entrada.

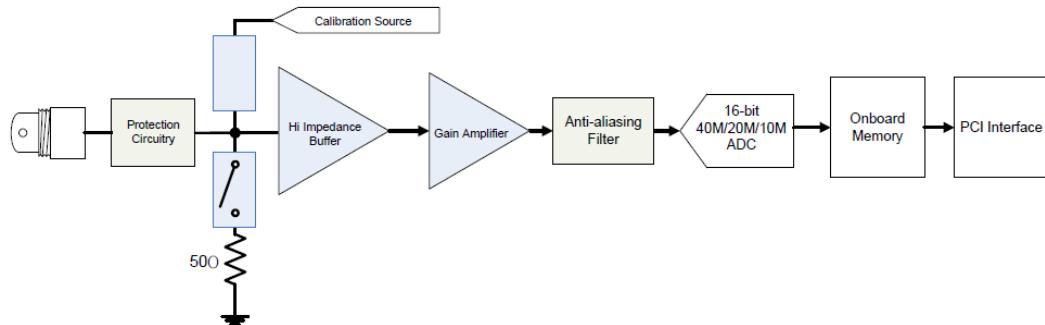


Figura 5.24: Diagrama de bloques del circuito de entrada de un canal.

El fabricante pone a disposición un driver que implementa una API para su configuración y manejo. Los detalles del driver, de la API y de sus principales funciones están descritos en el apéndice D.5.

5.1.2.1. Capacidad de transferencia.

El conversor dispone de una memoria interna de 512MB de SDRAM organizada en dos bancos. Las muestras digitalizadas se almacenan en esta memoria hasta que son transferidas al PC a través del bus PCI mediante un controlador DMA.

Cada uno de los bancos de memoria dispone de un único bus de acceso y, por lo tanto, solamente un dispositivo (el *ADC* para almacenar muestras o el controlador DMA para transferirlas) puede acceder simultáneamente a cada uno de ellos. El controlador PCI del *ADC* soporta una frecuencia máxima de reloj de 33MHz. A través de sus 32 líneas de datos es capaz de transferir 2 muestras de 16 bits por cada ciclo de bus.

Dispone de un bus de direccionamiento de 32 bits que le permite realizar transacciones de hasta 4GB en una única transferencia DMA. Además incorpora el protocolo *scatter-gather DMA* que facilita que la aplicación receptora de los datos no tenga que disponer de un único bloque de memoria de 4GB, permitiendo que dicho bloque se encuentre paginado en distintas zonas de la memoria.

El *ADC* debe usar una de las dos posibles estrategias que tiene implementadas para transferir las muestras capturadas hasta la memoria de la aplicación receptora de los datos (véase 5.2). Ambas estrategias se denominan modo de captura simple y modo de captura continuo y están descritas en los apéndices D.3 y D.4. El *ADC* funciona actualmente en modo continuo.

5.1.3. Tarjeta de inicio de captura.

La *tarjeta de inicio de captura* es la encargada de generar un pulso de disparo al *ADC* y al motor de elevación cuando el motor de rotación orienta el espejo de exploración

con un ángulo ϕ determinado. Este pulso es único en todo el proceso de captura y no se repite aunque el espejo vuelva a la posición inicial fijada en vueltas sucesivas.

La tarjeta, mostrada en la figura 5.25, se ha fabricado sobre un sustrato *FR4 (Flame Retardant 4)* de 1.6 milímetros de espesor. Ambas caras del dieléctrico están recubiertas por una capa de cobre de 35 μm . La tarjeta dispone de múltiples conectores que le permiten interactuar con cada uno de los elementos del sistema.

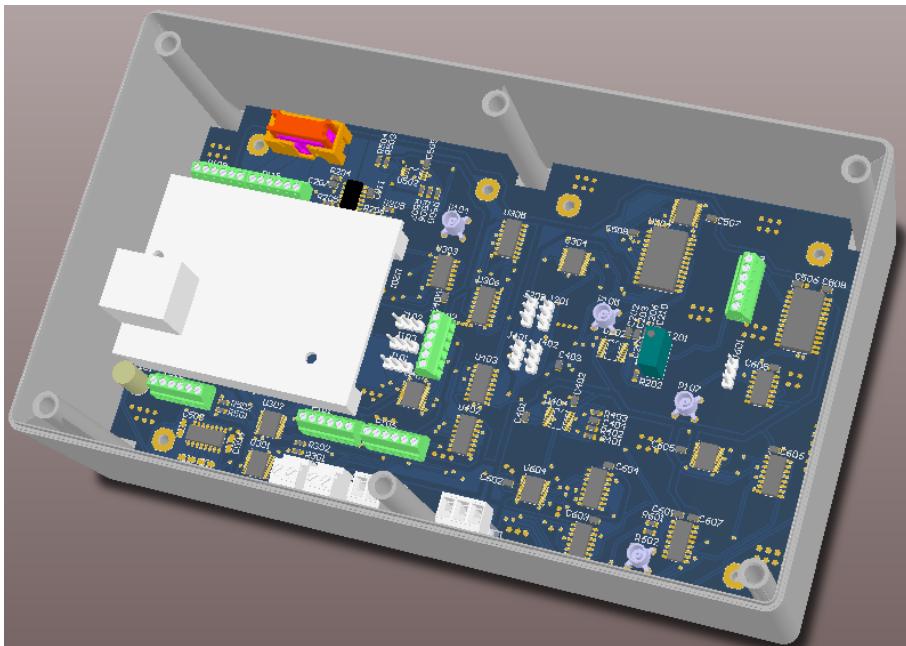


Figura 5.25: Tarjeta de inicio de captura.

En la siguiente lista se detalla cada uno de dichos elementos y la denominación de su correspondiente conector según los esquemas circuitales del apéndice E.1.

- Encoder del motor de rotación (P109).
- Encoder del motor de elevación (P101).
- Entradas/Salidas digitales del controlador del motor de rotación (P112 Y P113).
- Entradas/Salidas digitales del controlador del motor de elevación (P102).
- Referencias inductivas de los fines de carrera del motor de elevación (P106).
- Referencia inductiva del motor de rotación (P106).
- Señal *marker* del *DDS* (P105).
- Señal *trigger* del *DDS* (P104).
- Señal *trigger* del *ADC* (P107).

La tarjeta de inicio de captura incorpora un módulo MOD5270 (véase B.1), 2 *PLL* (*Phase Loop Lock*), un receptor de linea de modo diferencial, transceptores de bus para comunicar las entradas y salidas de los controladores con el microcontrolador, un puerto de serie para la comunicación con el controlador del motor de elevación, básculas tipo JK y contadores binarios de 4 bits asociados en cascada.

La tarjeta de inicio de captura tiene que trabajar con interfaces de distinta naturaleza por lo que requiere ser polarizada con tensiones de 3.3, 5 y 24V. El hardware desarrollado no se utiliza en su totalidad ya que muchas partes del diseño son diferentes alternativas para implementar una misma solución. Por ejemplo, los *PLL* han sido diseñados para realizar una gestión servomotor del motor de rotación de forma que los pulsos de su encoder estuvieran sincronizados con la señal de reloj del sistema (véase 5.1.1.1). Dicha solución no ha sido implementada en el sistema de sincronismo.

La tarjeta de inicio de captura recibe los pulsos del encoder del motor de rotación y convierte las señales diferenciales entregadas a modo común. La señal del encoder correspondiente al canal A sirve de señal de reloj a un contador de 12 bits formado por 3 contadores de 4 bits en cascada. En una vuelta del *espejo de exploración* el canal A genera 4000 pulsos. El contador de 12 bits permite un valor máximo de cuenta de 4096 unidades. El contador está gobernado por unas básculas tipo JK que determinan el inicio de la cuenta y el final de la misma.

La aplicación del sistema envía un mensaje UDP al MOD5270 en el que determina el valor de cuenta que ha de cargarse en el contador de 12 bits y éstearma la báscula JK de arranque. Esta báscula se dispara cuando se activa la referencia inductiva instalada en el espejo de exploración (véase el apartado 3.2.4.1).

Los contadores son programados de forma que cuando se reciban los pulsos deseados el valor de la cuenta desborde generando una señal RCO (Ripple Carry Out). Esta misma señal se propaga hasta el *ADC* y hasta el controlador del motor de elevación para iniciar la captura de muestras y el movimiento de barrido vertical respectivamente. El número de pulsos necesario para posicionar el espejo según la orientación $\phi=0$ desde su paso por la referencia inductiva es de 1430 pulsos.

En la figura 5.26 se muestra el funcionamiento de la señal de disparo para un valor de cuenta de 5 pulsos. En trazo verde se representa la referencia del captador de inducción. En color morado se representan los pulsos del canal A del encoder del motor de rotación. En color amarillo se representa la señal de disparo de inicio de captura. Por último, en trazo de color rosa se muestra el instante de inicio del movimiento del motor de elevación.



Figura 5.26: Captura de osciloscopio de pulsos de contadores y señal de disparo.

5.2. Subsistema de control.

Para que el usuario final pueda configurar y operar el sistema se ha desarrollado una GUI (Graphic User Interface) en Visual C++. Dicha aplicación se denomina **aplicación del sistema**. En la tabla 5.2 se muestran todos los sistemas con los cuales la aplicación interactúa.

Sistemas soportados
Sistema de polarización (en desarrollo).
Tarjeta de generación de señales de reloj.
Sistema de generación de la señal de radiofrecuencia.
Tarjeta de digitalización de la señal recibida.
Subsistema de apuntamiento.
Subsistema de sincronismo.

Tabla 5.2: Sistemas configurables desde la aplicación del sistema.

La aplicación hace uso de las APIs facilitadas por Maxon y ADLINK para gestionar el motor de rotación y la tarjeta *ADC* respectivamente. Parte del código de la aplicación se ha desarrollado utilizando el lenguaje *CLR* (*Common Language Runtime*) y otra parte mediante el lenguaje *ANSI* (American National Standards Institute) en su versión C11. La aplicación tiene un formulario principal, mostrado en la figura 5.27, desde donde el operador puede iniciar una captura de la escena a explorar.

Una vez finalizada la captura la aplicación almacena las muestras capturadas en un archivo. El usuario puede modificar el nombre con el cual se generará dicho archivo modificando los controles que se representan en la parte inferior izquierda de la figura.

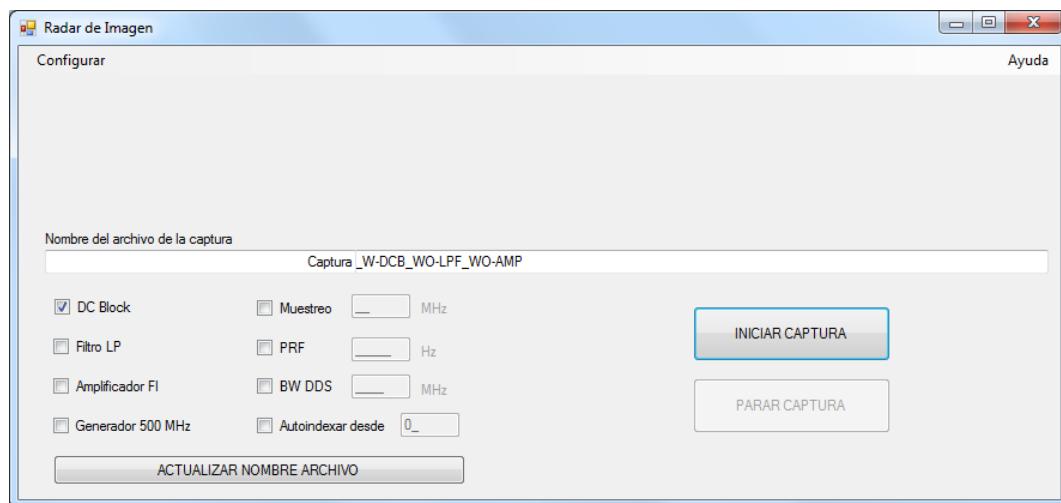


Figura 5.27: Formulario principal.

La opción de autoindexado permite generar automáticamente nombres de archivo con un sufijo numérico que se autoincrementa cada vez que se realiza una captura. En la parte superior del formulario se accede a un menu desplegable que permite acceder a la configuración de cada uno de los sistemas.

5.2.1. Configuración del ADC.

El formulario de configuración del *ADC* (figura 5.28) incorpora 4 pestañas. La pestaña identificada como *TARJETA* permite seleccionar el modelo de tarjeta *ADC* y el número que identifica la tarjeta.

La pestaña *CONFIGURACION CAPTURA* (figura 5.29) permite configurar los distintos parámetros de captura del *ADC* tales como el número de canales que se van a capturar, su impedancia de entrada, el origen de la señal de disparo (disparo mediante software o mediante señal externa), la fuente de reloj (reloj interno de la tarjeta o referencia externa), el submuestreo, el rango de entrada en tensión y el número de muestras a capturar. Este último parámetro es una variable del programa que sirve para detener el proceso de captura cuando se ha alcanzado el número de muestras requerido.

La pestaña *DESTINO CAPTURA* sirve para especificar la ubicación donde se almacenará el archivo con las muestras capturadas. La pestaña *CALIBRACIÓN* incorpora un botón de comando que al pulsarlo ejecuta un proceso de calibración del *ADC* (véase 5.1.2).

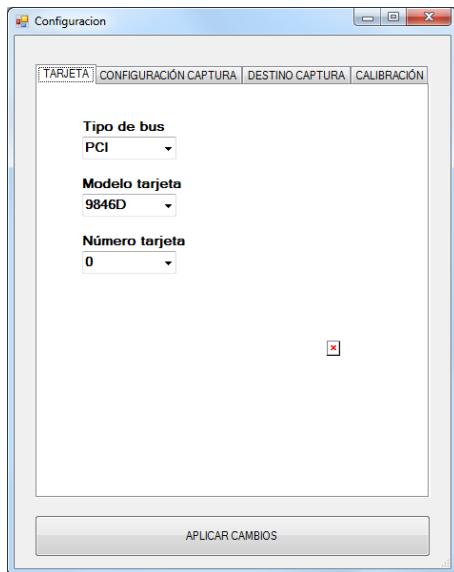


Figura 5.28: Formulario de configuración del ADC.

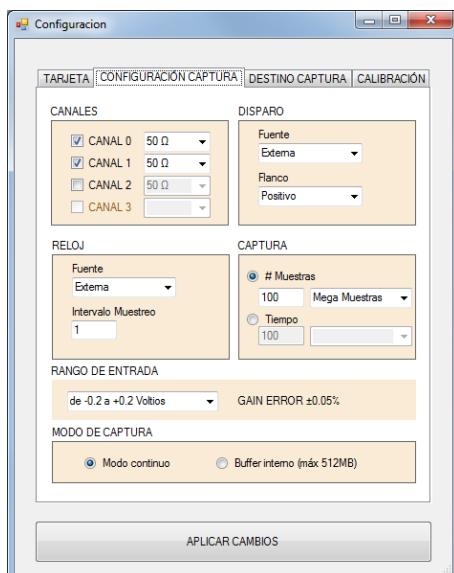


Figura 5.29: Formulario de configuración del ADC.

5.2.2. Configuración del motor de rotación.

La aplicación incorpora un formulario que permite controlar y configurar el motor de rotación. Dicho formulario se muestra en la figura 5.30

En la parte intermedia del formulario se dispone de un comando de opción que permite seleccionar una de las 3 posibles opciones. La primera de ellas, *No activar rotación*, indica a la aplicación que durante el proceso de captura no se ha de gestionar dicho motor de rotación.

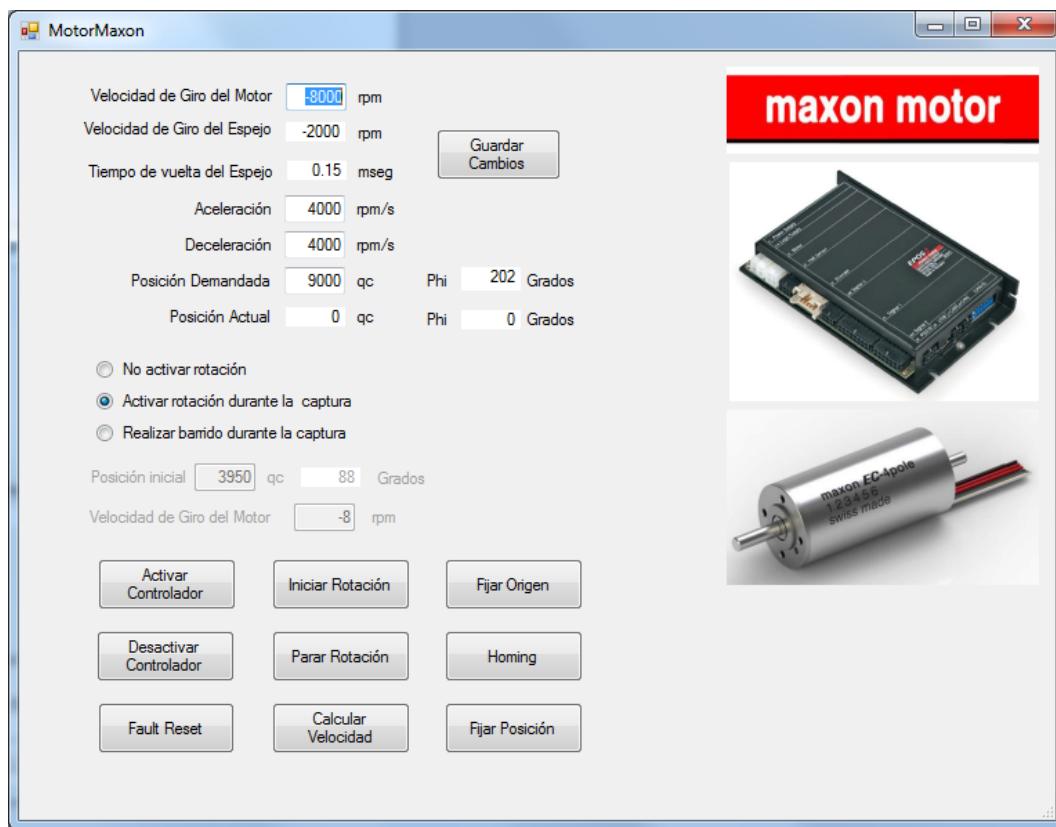


Figura 5.30: Formulario de control y configuración del motor de rotación.

La segunda opción obliga a la aplicación a que el motor rote a velocidad constante antes de comenzar con el proceso de captura. Los parámetros que determinan el movimiento descrito por el motor en esta opción se representan en la parte superior del formulario. El signo del parámetro que determina la velocidad establece el sentido de giro del motor. En un proceso de captura normal dicho valor debe ser negativo.

La tercera opción logra que el motor se posicione inicialmente en una posición determinada. Cuando comienza la captura, el motor comienza a girar conforme a la velocidad especificada debajo de dicha opción. Cuando la captura finaliza el motor se detiene. Además, en la parte inferior del formulario se dispone de una serie de comandos que permiten interactuar con el controlador del motor cuando éstos son pulsados.

El comando *Activar Controlador* hace que el controlador excite los bobinados del motor (esto no quiere decir que el motor gire). El comando *Desactivar Controlador* hace lo contrario. El comando *Fault Reset* sirve para anular cualquier alarma que se haya generado en el controlador.

El comando *Iniciar Rotación* logra que el motor comience a girar con la aceleración determinada en la parte superior del formulario hasta alcanzar la velocidad determinada en el mismo lugar. El comando *Parar Rotación* detiene el giro aplicando la deceleración establecida en la parte superior del formulario.

El comando *Calcular Velocidad* consulta al controlador la velocidad de giro actual en un bucle que se repite 100 veces. Tras finalizar el bucle se muestra en un mensaje emergente la velocidad promedio obtenida. El comando *Fijar Origen* indica al controlador del motor que, para la posición actual, el valor qc ha de ser 0.

El comando *Homing* inicia el proceso de *homming* (véase el apartado A.2.4). El comando *Fijar Posición* indica al controlador que posicione el motor con el valor qc definido como posición inicial en el formulario.

5.2.3. Configuración del motor de elevación.

El formulario mostrado en la figura 5.31 permite configurar y controlar el motor de elevación.

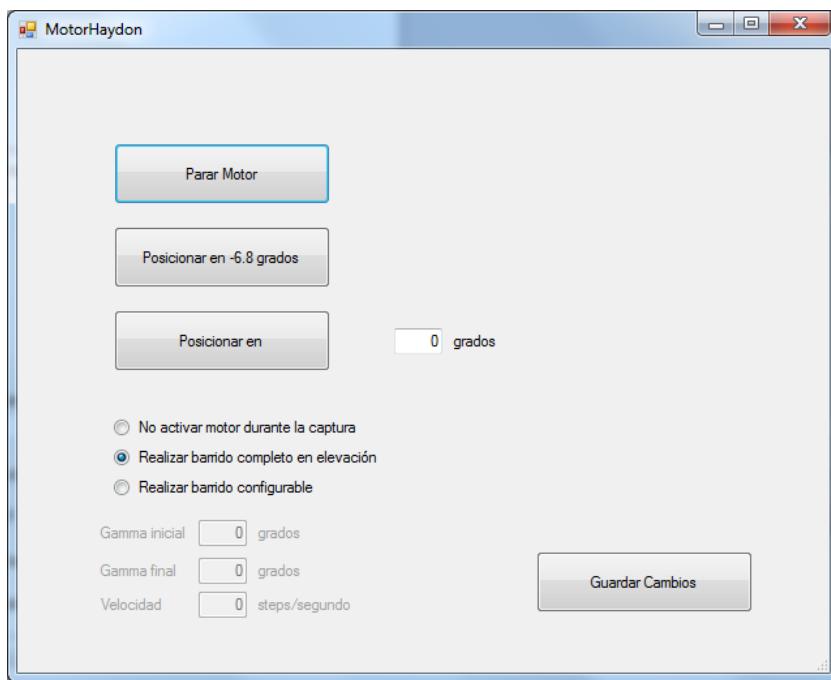


Figura 5.31: Formulario de control y configuración del motor de elevación.

Al igual que ocurre con el motor de rotación, se debe seleccionar una de las tres opciones disponibles para determinar la forma de interactuar del motor de elevación en el proceso de captura. La opción *No activar motor durante la captura* hace que el programa no gestione dicho motor durante la captura.

La opción *Realizar barrido completo* establece que, antes de que comience el proceso de captura, el programa posicione el motor de modo que el valor del ángulo γ sea de -6.8 grados. En el momento en el cual comienza la captura el motor inicia un movimiento de barrido vertical que finaliza cuando el ángulo elevación γ toma el valor +6.8 grados.

La opción *Realizar un barrido configurable* permite que el barrido descrito en el párrafo anterior tenga una posición de inicio, una posición final y una velocidad de barrido determinadas por los parámetros determinados en la parte inferior del formulario.

El comando *Para Motor* detiene cualquier maniobra que este realizando el controlador del motor de elevación. El comando *Posicionar en -6.8 grados* envía un mensaje al controlador en el cual se le indica que ejecute el programa que tiene albergado en memoria. Dicho programa ejecuta un movimiento de extensión del husillo hasta detectar el fin de carrera inferior (véase el apartado 3.2.4.2). Posteriormente ejecuta una maniobra de retracción del husillo que posiciona el motor de forma que el ángulo de elevación γ toma el valor -6.8 grados.

El comando *Posicionar en* ejecuta el proceso descrito en el párrafo anterior y posteriormente ejecuta una extensión del husillo que logra que el ángulo de elevación γ tome el valor indicado a la derecha de dicho comando.

5.2.4. Configuración del generador de señal *trigger*.

La aplicación dispone de un último formulario donde se puede configurar una señal de trigger externo. El formulario integra las funciones descritas en el apéndice C.1.

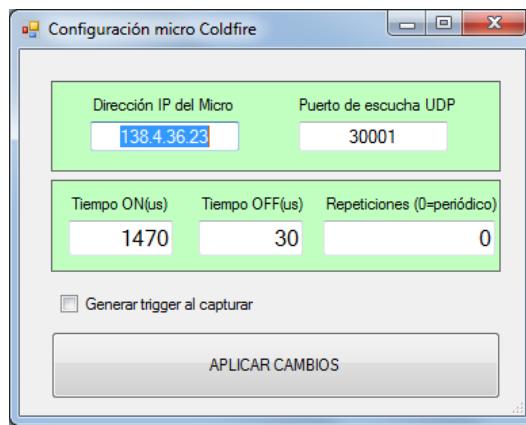


Figura 5.32: Formulario de configuración de la señal *trigger*.

En el instante previo al inicio de la captura la aplicación envía un mensaje al generador de trigger con la configuración establecida.

Capítulo 6

Mejoras propuestas.

Tras realizar un análisis de mayor profundidad, gracias a un mejor conocimiento de las características de los distintos equipos, hay que concluir que el sistema desarrollado no es el más óptimo para la explotación del equipo. Esto es debido a que el sistema de sincronismo y control desarrollado se basa en suponer una velocidad de los motores constante. En contra de lo supuesto inicialmente, el controlador del motor de rotación no es capaz de mantener la velocidad constante requerida (véase el apartado 3.2.2.4). Debido a esta falta de control en la velocidad, la estimación del ángulo ϕ para una determinada muestra incorpora un error que desplaza el posicionamiento de la muestra haciendo que las imágenes obtenidas no sean todo lo nítidas que podrían llegar a ser. Para evitar este fenómeno es necesario asociar unívocamente los pulsos del encoder del motor de rotación que determinan el ángulo ϕ con las muestras digitalizadas.

Una posible opción sería la de capturar en un tercer canal del *ADC* los pulsos del encoder, sin embargo, esta opción requeriría demasiado procesamiento de la señal capturada y además se debería reducir la frecuencia de muestreo del *ADC* para poder transferir a tiempo los datos a través del bus PCI (véase el apéndice D.4). La solución propuesta es similar a la utilizada para la detección del ángulo $\phi=0$ en sentido inverso. En vez de generar un pulso cuando se alcanza un valor de cuenta determinado, los contadores registran el número de pulsos del encoder detectados entre dos señales marker consecutivas. Los datos pueden ser almacenados en la memoria RAM del mismo módulo MOD5270 que programa actualmente los contadores de la tarjeta de inicio de captura.

Una vez procesadas las muestras capturadas se determina cuáles son las rampas en las que se ha detectado un blanco. Con dichos datos, la aplicación envía un mensaje UDP al módulo en el que solicita la conversión correspondiente entre número de rampa el valor angulares ϕ . De esta manera desaparece el término de error que aparece cuando el motor gira a velocidad no constante. Para implementar esta solución es necesario dimensionar los contadores, el espacio de memoria RAM requerido y el conocer el tiempo disponible entre rampa y rampa para la lectura de los contadores y su almacenamiento en memoria. Con los parámetros de operación utilizados actualmente se genera una señal marker cada 150 μs . En dicho lapso de tiempo y con una velocidad de giro del

espejo de exploración de 2,000 rpm se generan 80 qc's (véase 3.2.2.2) entre dos señales marker. Durante la captura, cuya duración es de 1.17 s (tiempo necesario para barrer verticalmente desde -6.8° hasta $+6.8^\circ$ cuando el motor de elevación se mueve a una velocidad de 827 steps/s), se generan 7,800 señales de *marker*.

El contador debe tener un valor máximo de cuenta que permita registrar todos los pulsos generados por el encoder entre 2 rampas sin desbordarse. Por este motivo los contadores deben tener una capacidad cuenta de 8 bits para los parámetros de captura fijados. Para almacenar los 7,800 intervalos producidos durante la captura se requiere una memoria RAM de 6.24 KB que es posible reservar dentro de los 64KB que hay disponibles en el microcontrolador (además de los 8 MB SDRAM y los 512 KB FLASH disponibles en el módulo). Cada vez que se detecta una señal *marker* se produce una interrupción en el microcontrolador. En la rutina de atención de la interrupción se incrementa una variable denominada *numeroDeRampa*, se lee el valor de la cuenta y se resta el valor obtenido respecto del anterior valor registrado. De esta forma se obtiene el número de pulsos registrado entre 2 señales *marker* sin necesidad de reiniciar ningún contador evitando que durante dicha orden de reset se dejen de registrar pulsos.

Hay que tener en cuenta que el error en la determinación del ángulo causado por la pérdida de pulsos del encoder (o por fallos de éste) es acumulativo ya que el encoder es relativo. Para tratar de minimizar este problema también se propone capturar cada pulso de la señal index del encoder. Dicha señal genera un total de 4 pulsos por vuelta. El tiempo requerido para el tratamiento de la interrupción se prevé inferior a los 500 ns. Aunque el motor de elevación trabaja a una velocidad 3 órdenes de magnitud inferior y, por ello, es menos sensible a este problema dicha solución propuesta también puede implementarse con el encoder del motor de elevación.

Otra posible mejora puede ser la de disminuir el tiempo de captura para obtener una mayor frecuencia de refresco entre imágenes. El tiempo de captura viene determinado por la máxima velocidad de rotación. El sistema actual tiene diversas limitaciones a este respecto que han de ser subsanadas. En primer lugar, el espejo de exploración no está equilibrado; Ello provoca que a velocidades mayores se produzcan fuertes vibraciones en el sistema de antenas según se describe en el apartado 3.2.2.4. Se puede realizar un mecanizado de vaciado que equilibre la pieza.

En segundo lugar, el encoder del motor de rotación no soporta velocidades mayores a 12,000 rpm y limita la velocidad máxima del motor que es de 16,700 rpm. A 12,000 rpm el tiempo de captura es de 0.78 s. Es posible sustituir el encoder actual por otro cuya resolución es la mitad (500 pulsos/vuelta) pero suficiente y tolera velocidades de hasta 24,000 rpm. A partir de 12,500 rpm el sistema está limitado por el motor de elevación. A dicha velocidad el tiempo de captura es de 0.75 s; Para mayores velocidades es necesario sustituir el motor de elevación por otro cuyo paso sea superior. Existe margen de maniobra para modificar dicho parámetro sin que afecte significativamente al sistema. En tal caso, con el motor actual a 16,700 rpm el tiempo de captura es de 0.56s.

Para lograr mayores velocidades se debe prescindir de la reductora. Actualmente el espejo de exploración desequilibrado impone un par de 235 mNm. La reductora tiene una eficiencia del 96 % y por lo tanto si se elimina ésta el par que observará el motor con el disco actual es de 225 mNm. El motor actual soporta un par nominal de 124 mNm y un par máximo 3,220 mNm por lo que se debería, además de equilibrar el disco, tratar de disminuir la masa de éste. El tiempo de captura con el motor trabajando a 16,700 rpm y sin reductora es de 0.14 segundos.

Otra posible mejora es la de atenuar la señal generada por el DDS mediante un control digital de forma que sea posible eliminar el atenuador manual descrito en el apartado 5.1.

Capítulo 7

Conclusiones.

El escáner es hoy un instrumento operativo con una velocidad de refresco de una imagen por segundo. Con el sistema actual es posible duplicar dicha velocidad. La limitación en la velocidad de refresco se debe a restricciones de la mecánica del espejo de exploración. Existen varias soluciones para salvar dichas restricciones que se describen en el capítulo 6.

La aportación fundamental, además de la integración, ha sido el desarrollo de los subsistemas de control y de sincronismo.

Bibliografía

- [1] Beatriz Mencia-Oliva, Jesús Grajal, Omar A. Yeste-Ojeda, Gorka Rubio-Cidre, and Alejandro Badolato, "Low-Cost CW-LFM Radar Sensor at 100 GHz", IEEE Transactions on Microwave Theory and Techniques, VOL. 61, NO. 2, FEBRUARY 2013
- [2] S. Nicolson, K. Yau, S. Pruvost, V. Danelon, P. Chevalier, P. Garcia, A. Chantre, B. Sautreuil, and S. Voinigescu, "A low-voltage SiGe BiCMOS 77-GHz automotive radar chipset," IEEE Transactions on Microwave Theory and Techniques, vol. 56, no. 5, pp. 1092–1104, May 2008.
- [3] J. Hasch, E. Topak, R. Schnabel, T. Zwick, R. Weigel, and C. Waldschmidt, "Millimeter-wave technology for automotive radar sensors in the 77 GHz frequency band," IEEE Trans. Microw. Theory Techn., vol. 60, no. 3, pp. 845–860 Mar. 2012.
- [4] W. Johannes, H. Essen, S. Stanko, R. Sommer, A. Wahlen, J. Wilcke, C. Wagner, M. Schlechtweg, and A. Tessmann, "Miniaturized high resolution synthetic aperture radar at 94 GHz for microlite aircraft or UAV," in IEEE Sensors, Oct. 2011, pp. 2022–2025.
- [5] B. Mencia-Oliva, J. Grajal, and A. Badolato, "100-GHz FMCW radar front-end for ISAR and 3-D imaging," in IEEE Radar Conf., May 2011, pp. 389–392.
- [6] M. Soumekh, Synthetic Aperture Radar Signal Processing With MATLAB Algorithms. New York: Wiley, 1999.
- [7] R. M. M. G. Carrara and R. S. Goodman, Spotlight Synthetic Aperture Radar Signal Processing Algorithms. New York: Wiley, 1999.
- [8] R. Appleby and H. Wallace, "Standoff detection of weapons and contraband in the 100 GHz to 1 THz region," IEEE Trans. Antennas Propag., vol. 55, no. 11, pp. 2944–2956, Nov. 2007

- [9] K. Cooper, R. Dengler, N. Llombart, B. Thomas, G. Chattopadhyay, and P. Siegel, "THz imaging radar for standoff personnel screening," IEEE Trans. Terahertz Sci. Technol., vol. 1, no. 1, pp. 169–182, Sep. 2011.
- [10] Assessment of Millimeter-Wave and Terahertz Technology for Detection and Identification of Concealed Explosives and Weapons. ISBN 9780309104692
- [11] Equipo de detección de rayos X RapidScan Secure-1000.
<http://www.rapiscansystems.com/>
- [12] Equipo de detección de rayos X Secure-1000.
<http://www.tek84.com/>
- [13] Equipo de detección milimétrico ProVision-2.
<http://www.safeviewinc.com/advancedimaging/provision-2.htm>
- [14] Position statement of the Health Physics Society
http://hps.org/documents/securityscreening_ps017-1.pdf
- [15] Informe de la FAA (Federal Aviation Administration)
http://www.faa.gov/data_research/research/med_humanfacs/aeromedical/radiobiology/cari6/ine
- [16] Informe de la agencia de protección ambiental
<http://www.epa.gov/radtown/cosmic.html>
- [17] Informe técnico de la TSA (Transportation Security Administration) sobre escáner de rayos X.
http://www.tsa.gov/video/pdfs/survey_ait_032213/CMH_S50952004_Secure1000_2012-11-15.pdf
- [18] Informe de la Comisión Europea
Scientific Committee on Emerging and Newly Identified Health Risks
http://ec.europa.eu/health/scientific_committees/emerging/docs/scenihr_o_036.pdf
- [19] Artículo de la organización Propublica
<http://www.propublica.org/article/tsa-removes-x-ray-body-scanners-from-major-airports>
- [20] Linear Technology Application Note 32
High Efficiency Linear Regulators
Jim Williams 1989
<http://cds.linear.com/docs/en/application-note/an32f.pdf>
- [21] Garcia-Pino, A.; Gonzalez-Valdes, B.; Mencia-Oliva, B.; de la Fuente, J.G.; Lopez, O.R.; Sanmartin, J.L.B., "Bifocal Reflector antenna system for radar imaging at 300 GHz," *Antennas and Propagation (EuCAP), 2013 7th European Conference on*, vol., no., pp.3814,3818, 8-12 April 2013

- [22] Understanding Thermal Dissipation and Design of a Heatsink <http://www.ti.com/lit/an/slva462/slva462.pdf>
- [23] P. Huggard, M. Oldfield, B. Moyna, B. Ellison, D. Matheson, A. Bennett, C. Gaffard, T. Oakley, and J. Nash "94 GHz FMCW cloud radar". In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, Oct. 2008
- [24] M. Guldogan, F.. Gustafsson, U. Orguner, S. Bjorklund, H.Petersson, and A. Nezirovic. "Human gait parameter estimation based on micro-Doppler signatures using particle filters". In *Acoustics, speed and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pp. 5940–5943, may 2011.
- [25] G. BrooKer, J. Martinez, and R. Hennessey. "Milimetre wave radar imaging of mining vehicles". In *Radar Conference (EuRAD) 2010 European*, pp. 284–287,30 2010-oct. 1 2010
- [26] Rangwala, M.; Feinian Wang; Sarabandi, K., "Study of millimeter-wave radar for helicopter assisted landing system" Geoscience and Remote Sensing Symposium, 2007. IGARSS 2007. IEEE International , vol., no., pp.777,780, 23-28 July 2007
- [27] W. Johannes, H. Essen, S. Stanko, R. Sommer, A. Wahlen, J. Wilcke, C. Wagner, M. Schlechtweg, and A. Tessmann. "Miniaturized high resolution Synthetic Aperture Radar ar 94 GHz- for microlite aircraft or UAV". In *Sensors, 2011 IEEE*, pp. 2022–2025, oct. 2011.
- [28] K. Van Caejenberghe, K. Brakora, and K. Sarabandi. "A 94 GHz OFDM Frequency Scanning Radar for Autonomous Landing Guidance". In *Radar Conference, 2007 IEEE*, pp. 248–153, april 2007
- [29] H. Essen, O. Konrad, A. Wahlen, K. Schilz, K. Jager, and M. Hebel. "ISAR imaging of helicopters using milimeter wave radars" In *International Journal of Microwave and Wireless Technologies*, 2009
- [30] S. Hantscher, B. Schlenther, M. Hagelen, S. Lang, H. Hessen, A. Tessmann, A. Huls-mann, P. Leuther, and M. Schlechtweg. "Security Pre-screening of Moving Persons Using a Rotating Multichannel W-Band Radar". *Microwave Theory and Techniques, IEEE Transactions on*, Vol. 60, No. 3, pp. 870–880, march 2012
- [31] B. Kapilevich, Y. Pinhasi, M. Anisimov, B. Litvak, and D. Hardon. "FMCW MM-wave non-imaging sensor for detecting hidden objects". In *Microwave Workshop Series on Millimeter Wave Integration Technologies (IMWS), 2011 IEEE MTT-S International*, pp. 101–104, sept. 2011
- [32] S. Gu, C. Li, X. Gao, Z. Sun, and G. Fang, "Terahertz Aperture Synthesized Imaging With Fan-Beam Scanning for Personnel Screening," *Microwave Theory and Techniques, IEEE Transactions on* , vol.60, no.12, pp.3877,3885, Dec. 2012

- [33] Qian Song, Yuejin Zhao, Albert Redo-Sanchez, Cunlin Zhang, Xiaohua Liu, "Fast continuous terahertz wave imaging system for security", *Optics Communications*, Volume 282, Issue 10, 15 May 2009, pp 2019–2022
- [34] Thomas M. Goyette *a , Jason C. Dickinson A , Kurt J. Linden B , William R. Neal B , Cecil S. Joseph A , William J. Gorveatt A , Jerry Waldman A , Robert Giles A , William E. Nixon, "1.56 Terahertz 2-frames per second standoff imaging"
- [35] Cooper, K.B.; Trabert, J.F.; Dengler, R.J., "Carrier noise-limited penetration in THz radar imaging," *Microwave Symposium Digest (MTT), 2012 IEEE MTT-S International*, vol., no., pp.1,3, 17–22 June 2012
- [36] Sheen, D.M.; McMakin, D.L.; Hall, T.E.; Severtsen, R.H., "Active millimeter-wave standoff and portal imaging techniques for personnel screening," *Technologies for Homeland Security, 2009. HST '09. IEEE Conference on*, vol., no., pp.440,447, 11-12 May 2009
- [37] am Weg, Christian and von Spiegel, Wolff and Henneberger, Ralf and Zimmermann, Ralf and Loeffler, Torsten and Roskos, HartmutG., "Fast Active THz Cameras with Ranging Capabilities". *Journal of Infrared, Millimeter, and Terahertz Waves*

Apéndices

Apéndice A

Documentación del controlador del motor de rotación.

A.1. Modos de regulación.

A.1.1. Control de corriente.

El controlador, mediante un bucle de control de corriente (figura A.1) basado en un sistema PI (*Proportional and Integration controller*), es capaz de gestionar el motor de forma que éste entregue en su eje una fuerza de giro o torque constante. Para ello, el controlador muestrea la señal de retorno del lazo a una frecuencia de 10 KHz.

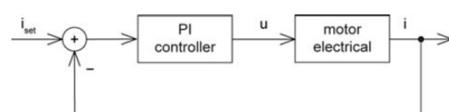


Figura A.1: Controlador PI de corriente.

A.1.2. Control de velocidad.

El controlador de velocidad (figura A.2) se apoya en el control de corriente anteriormente descrito para lograr una velocidad de giro del motor constante. Una vez establecida la velocidad de giro demandada un control de lazo cerrado, que muestrea la señal de retorno a una frecuencia de 1 KHz, trata de mantener dicha velocidad de forma constante.

Durante 1 ms se contabilizan los pulsos del encoder detectados y se calcula la velocidad actual de giro del motor para, posteriormente, introducir el valor obtenido en

A. Documentación del controlador del motor de rotación.

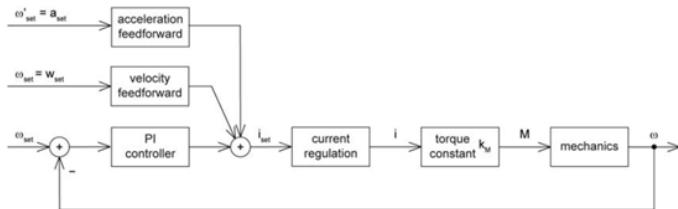


Figura A.2: Controlador PI de velocidad.

el control de lazo. El controlador puede promediar dicho valor antes de entregárselo al controlador de lazo. Para ello, implementa un filtro de primer orden que se rige por la ecuación en diferencias A.1 cuya función de transferencia se muestra en la ecuación A.2 y donde λ vale 2^{-5} .

$$y[k] = (1 - \lambda) \cdot y[k - 1] + \lambda \cdot x[k] \quad (\text{A.1})$$

$$H(Z) = \frac{Y(Z)}{X(Z)} = \frac{\lambda}{1 - (1 - \lambda) \cdot Z^{-1}} \quad (\text{A.2})$$

En la figura A.3 se muestra la respuesta del filtro en frecuencia. Para activar este filtrado hay que poner a 1 el bit número 3 del objeto del diccionario de objetos cuya dirección es 0x2008.

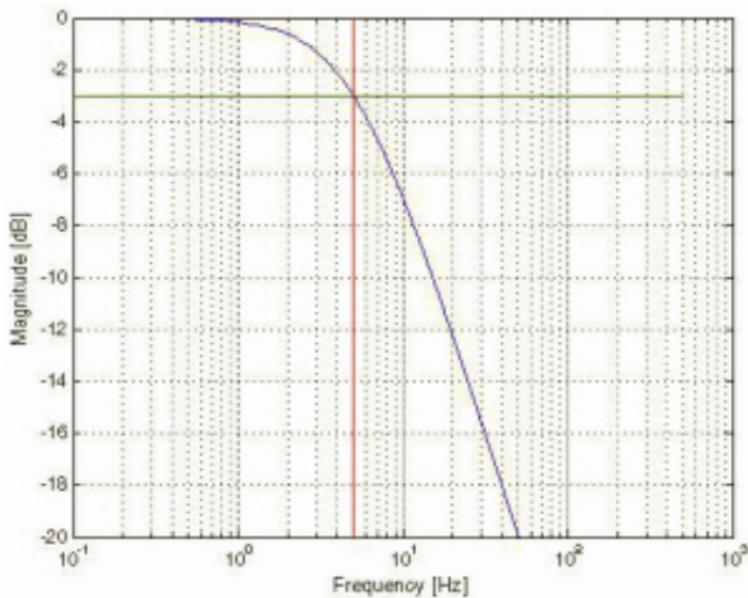


Figura A.3: Respuesta del filtro de promediado de velocidad.

La velocidad de giro demandada se puede determinar mediante programación o mediante una entrada analógica diferencial 0-5V cuya resolución es de 12 bits. En el

caso de controlarse la velocidad mediante una entrada analógica se ha de definir la velocidad de *offset* (velocidad demandada cuando la entrada tiene un potencial de 0V) y una constante de proporcionalidad K_w (con unidades $\frac{rad/s}{V}$).

Además, se pueden añadir al control de lazo cerrado dos entradas analógicas similares a la anterior mediante las cuales poder indicar al controlador la velocidad y aceleración previstas a muy corto plazo (*acceleration and velocity feedforward*) de forma que se antepongan a un posible error en la velocidad demandada debido a modificación previsible del torque en el tiempo.

A.1.3. Control de posición.

Mediante un controlador PID (*Proportional Integration and Derivation controller*) y apoyándose en el sistema de control de corriente es posible fijar una posición determinada del rotor. Al igual que el control de velocidad, el controlador de posición dispone de dos entradas de velocidad y aceleración previas que son capaces de anteponerse a variaciones en la posición debidas a la variación prevista del torque en el tiempo.

El controlador PID (figura A.4) muestrea las señales de retorno de lazo y las entradas analógicas a una frecuencia de 1KHz. Las unidades de posicionamiento que utiliza el controlador son qc's (descritas en 3.2.2.2).

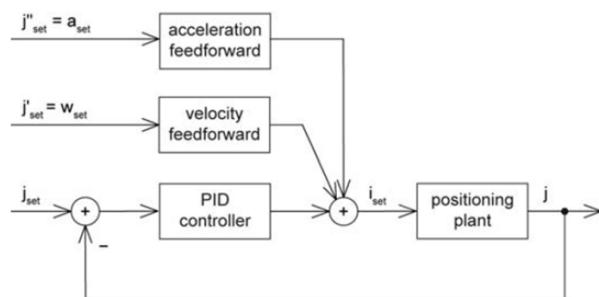


Figura A.4: Controlador PID de posición.

A.2. Control del motor de rotación mediante Epos Studio.

A.2.1. Introducción.

El motor de rotación se puede controlar de forma sencilla a través de un puerto USB haciendo uso de la *GUI* puesta a disposición por el fabricante en su página web (<http://www.maxonmotor.es>). Para localizar la aplicación hay que seleccionar el controlador EPOS2 70/10 y posteriormente en la sección de descargas seleccionar *Setup EPOS2 Studio 2.00*.

El puerto USB del PC debe proporcionar una corriente a 5V suficiente para que las comunicaciones del equipo funcionen correctamente. Se debe tener en cuenta que no todos los puertos USB (incluso de un mismo equipo) entregan la misma potencia.

Una vez instalada se debe ejecutar el asistente de configuración (*Startup Wizard*) para indicar al programa las características del motor y del encoder acoplado.

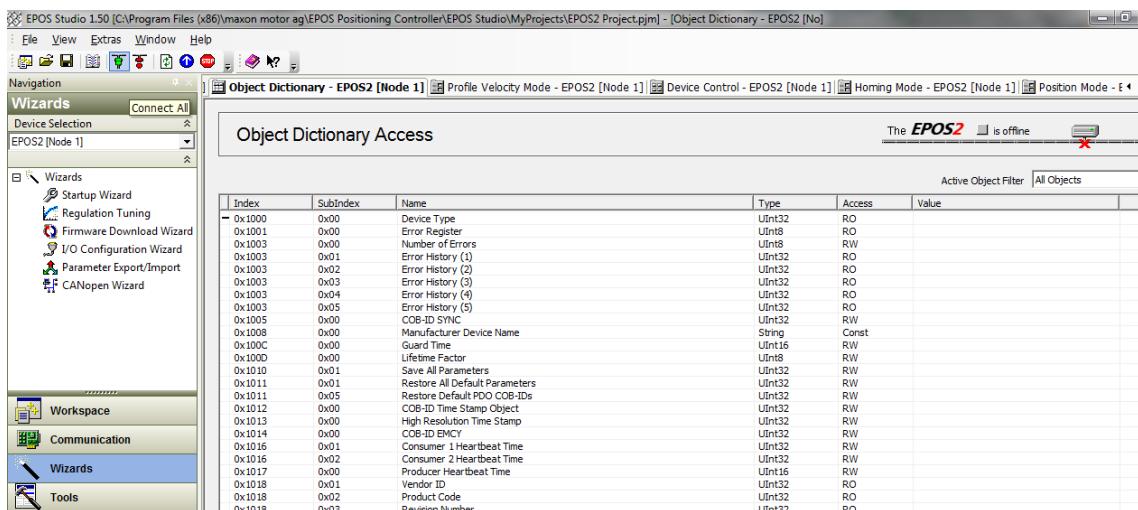


Figura A.5: *EPOS2 Studio 2.00*.

Tras el asistente de configuración se ha de pulsar el icono superior con luces verdes (*Connect all* véase la figura A.5) para comunicarse con la unidad.

A continuación, si es la primera vez que se inicializa el controlador, se debería ejecutar un proceso de calibrado (*Regulation Tuning* descrito en A.3) para ajustar los parámetros básicos de regulación.

En la parte superior de la figura A.5 hay una serie de pestañas que permiten acceder a las distintas secciones del programa. A continuación se describen las secciones más relevantes para un manejo básico del motor.

La pestaña Object Dictionary permite consultar y modificar el diccionario de objetos (véase el apartado 3.2.2.3).

A.2.2. *Profile Velocity Mode.*

La pestaña *Profile Velocity Mode* (figura A.6) permite controlar la velocidad de giro en unidades rpm. Se le puede indicar a la aplicación la aceleración máxima para alcanzar la velocidad objetivo y el perfil de aceleración.

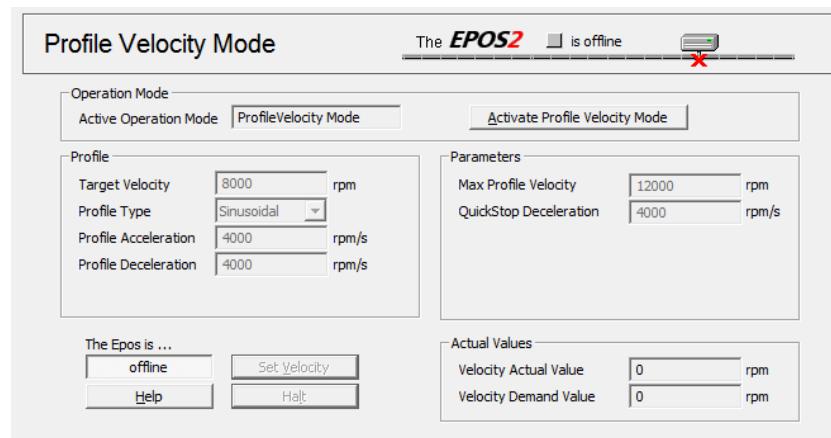


Figura A.6: Profile Velocity Mode.

Se puede elegir entre dos posibles perfiles de aceleración: trapezoidal y sinusoidal. En la figura A.7 se muestra el perfil de aceleración sinusoidal basado en la función \sin^2 y la evolución de la velocidad en el tiempo.

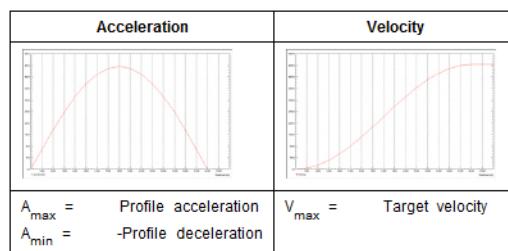


Figura A.7: Perfil de aceleración en \sin^2 .

Para poner en marcha este modo se debe pulsar en el botón de comando *Activate Profile Velocity Mode*; aunque para ello la unidad debe de estar previamente conectada (*Online*).

Para conectar la unidad hay que pulsar en el botón de comando que aparece en la parte inferior (figura A.6). Cuando la unidad está conecta bajo la etiqueta '*The Epos is...*' aparece *Disabled* en lugar de *Offline*.

En esta situación el controlador comienza a excitar los bobinados del motor tratando de que la velocidad del motor sea nula (ofreciendo resistencia si se trata de mover el espejo manualmente) ya que ésta es la velocidad demandada por defecto.

Para poner en marcha el motor hay que pulsar sobre el botón de comando *Set Velocity*. En la parte inferior de la figura A.6 se muestra la velocidad demandada por el controlador y la velocidad de rotación del motor.

Para detener el motor se debe pulsar en el botón de comando *Halt*.

A.2.3. Profile Position Mode.

La pestaña *Profile Position Mode* (figura A.8) permite controlar el posicionamiento del rotor en unidades qc (véase apartado 3.2.2.2).

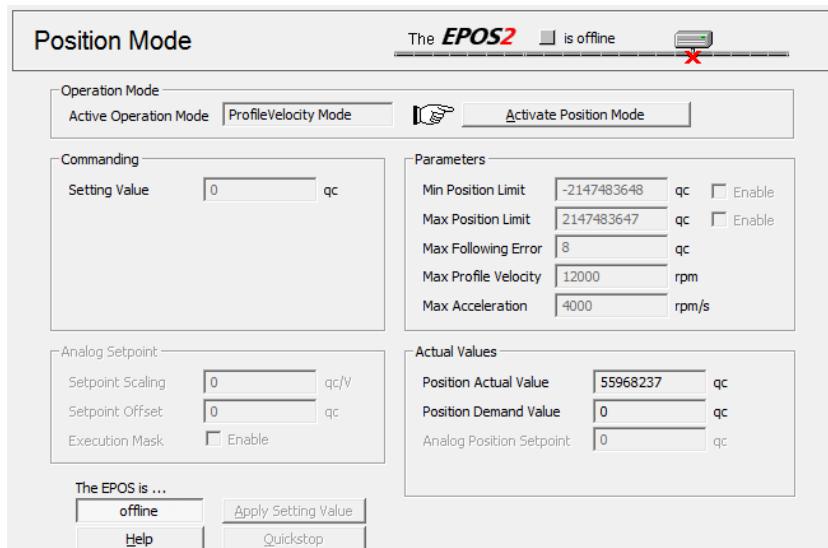


Figura A.8: Profile Position Mode.

Para poner en marcha este modo se debe pulsar en el botón de comando *Activate Profile Position Mode* aunque para ello la unidad debe de estar previamente conectada (véase A.2.2).

Cuando el controlador pasa a modo *Online* se asocia la posición del encoder en dicho momento de activación al valor absoluto qc que tiene cargado en memoria (véase A.2.4) y comienza a excitar los bobinados para mantener la misma posición (ofreciendo resistencia si se trata de mover el espejo manualmente).

En el panel *Parameters* se define el rango de las posiciones permitidas, el máximo error tolerable en la posición, la máxima velocidad de giro para alcanzar la nueva posición demandada y la máxima aceleración permitida para alcanzar dicha velocidad máxima.

Para provocar un movimiento del motor se establece la nueva posición en qc's y se pulsa en botón de comando *Apply Setting Value*.

A.2.4. Homing Mode.

La pestaña *Homing Mode* (figura A.9) permite definir el comportamiento del controlador en la búsqueda de una posible posición inicial requerida en el caso de que se quiera usar el motor mediante un control de posicionamiento (véase A.2.3).

La forma más precisa de ubicar el rotor en una determinada posición se logra mediante la lectura del canal *Index* del encoder (véase 3.2.2.2). Sin embargo, el espejo a posicionar (véase 3.1.4) está acoplado a una reductora 4:1 (3.2.2.1) y, por tanto, se obtienen hasta 4 referencias *Index* por cada vuelta del espejo.

Para resolver esta ambigüedad el controlador gira el rotor hasta detectar una referencia externa la cual genera un único pulso por cada vuelta del espejo (véase 3.2.4.1). Una vez detectada la referencia externa el controlador gira el rotor hasta detectar el primer pulso del canal *Index*.

A partir de esta posición se realiza una rotación adicional (definida en unidades qc) que sitúa el espejo en la posición inicial deseada. La posición alcanzada se denomina *Home*.

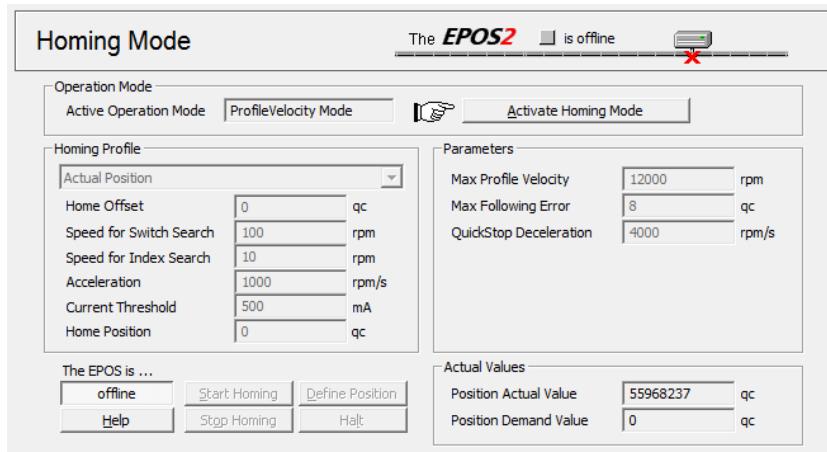


Figura A.9: Homing Mode.

En el panel *Parameters* se establece la máxima velocidad permitida para alcanzar la posición homing, el máximo error permitido en la determinación de la posición *home* y la máxima deceleración permitida.

En el panel *Homing Profile* se define la velocidad de giro para la detección de la referencia externa, la velocidad de giro para la detección del pulso *Index*, el desplazamiento adicional desde el pulso *Index* hasta la posición *Home* y la corriente con la cual se excitan los bobinados una vez alcanzada dicha posición para mantenerla fija.

Por último, es posible definir el valor qc de la memoria del controlador correspondiente a la posición *Home*. Una práctica habitual es poner a 0 dicho contador cuando se ha alcanzado la posición *Home*, de tal forma que, si se le indica al controlador que

se posicione en +8000 qc, el espejo rotará exactamente media vuelta desde la posición home.

A.2.5. Device Control.

La principal acción que se realiza en la pestaña *Device control* (figura A.10) es la de anular las alarmas producidas en el controlador.

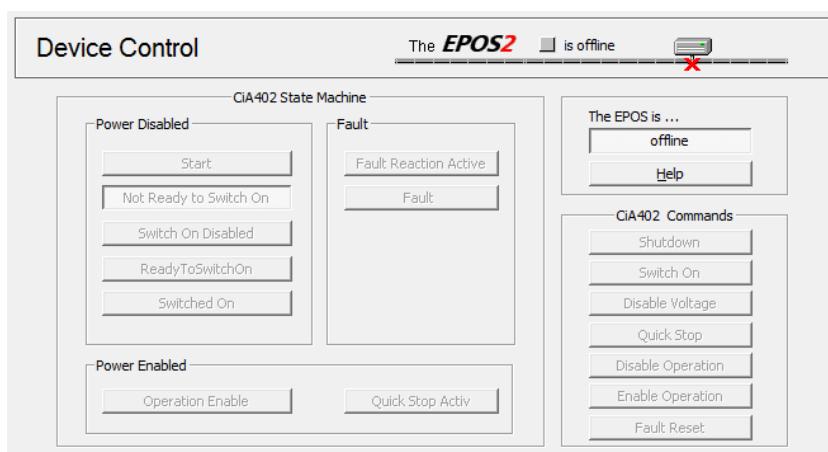


Figura A.10: Device Control.

Las alarmas pueden ser debidas a un insuficiente suministro de potencia al controlador por parte de la fuente de alimentación, a un exceso de potencia aplicado al motor (debida, por ejemplo, a una aceleración excesiva para la carga acoplada al rotor) o a fallos en la detección de las señales del encoder o de los sensores *Hall*.

Cuando se produce esta situación de alarma un led ubicado en el exterior de la carcasa del controlador se ilumina de color rojo. Para borrar dicha alarma se ha de pulsar en el botón de comando *Fault reset* ubicado en el panel *CIA403 Commands*.

A.3. Calibración del motor.

Existen dos modalidades de calibración: automática y modo experto. En la primera de ellas el software se encarga de realizar todos los ajustes sin intervención alguna por parte del usuario. En el modo experto la calibración se realiza en 4 etapas:

- Cascada.
- Identificación.
- Parametrización.

- Verificación.

En la etapa denominada cascada se le indica al controlador (figura A.11) cuál es la variable principal que gobernará la regulación (velocidad, posición o corriente). Posteriormente se pueden fijar manualmente las constantes del controlador PI o PID (según sea el caso) de dicha variable.

Las regulaciones en velocidad y en posición son sistemas que se apoyan a su vez en la regulación en corriente. Por ello, cuando se selecciona uno de estos modos de regulación, el software permite modificar también los parámetros PI del controlador de corriente.



Figura A.11: *Expert Tuning*: Etapa de cascada.

Posteriormente en la etapa de identificación (figura A.12) el controlador trata de conocer las propiedades iniciales del sistema acoplado al eje del motor. Ajustando un parámetro denominado amplitud, propiedades no lineales, como la fricción de Coulomb, pueden ser implementadas en el modelo del sistema o también denominado modelo de planta.



Figura A.12: *Expert Tuning*: Etapa de identificación.

La amplitud del sistema base se define en amperios mientras que la amplitud del sistema subrogado se mide en qc's. La etapa de identificación necesita realizar una rotación completa del cuerpo acoplado al eje de salida de la reductora para un correcto ajuste y, por lo tanto, es necesario fijar dicha amplitud en 16,000 qc's (vease 3.2.2.2).



Figura A.13: *Expert Tuning*: Etapa de parametrización.

En la etapa de parametrización (figura A.13) se terminan de definir las constantes de los controladores PI o PID (según sea el caso). Para ello, se puede controlar la

rigidez del ajuste de cada una de las variables de regulación. Una mayor rigidez (*hard*) provocará que el sistema reaccione de forma más brusca ante posibles cambios mientras que una menor rigidez (*soft*) amortiguará dichas variaciones.

Finalmente, en la etapa de verificación (figura A.14) el controlador realiza pruebas de regulación y registra gráficamente la evolución en el tiempo de la variable demandada y del valor real medido.

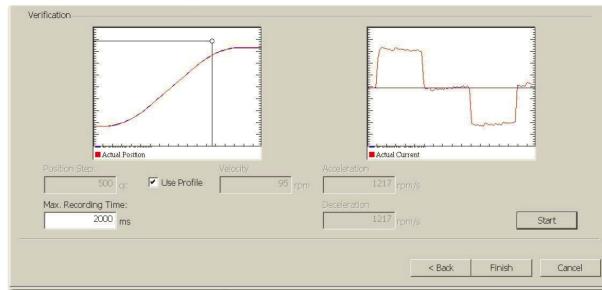


Figura A.14: *Expert Tuning*: Etapa de verificación.

Los parámetros susceptibles de calibración en ambos modos (automático y experto) son:

Parámetro	Descripción
<i>Current Regulator P-Gain</i>	Ganancia de proporcionalidad del controlador de corriente
<i>Current Regulator I-Gain</i>	Ganancia de integración del controlador de corriente
<i>Velocity Regulator P-Gain</i>	Ganancia de proporcionalidad del controlador de velocidad
<i>Velocity Regulator I-Gain</i>	Ganancia de integración del controlador de velocidad
<i>Velocity Feedforward Factor in Speed Regulator</i>	Factor de velocidad previo del controlador de velocidad
<i>Acceleration Feedforward Factor in Speed Regulator</i>	Factor de aceleración previo del controlador de velocidad
<i>Position Regulator P-Gain</i>	Ganancia de proporcionalidad del controlador de posición
<i>Position Regulator I-Gain</i>	Ganancia de integración del controlador de posición
<i>Position Regulator D-Gain</i>	Ganancia de derivación del controlador de posición
<i>Velocity Feed Forward Factor</i>	Factor de velocidad previo del controlador de posición
<i>Acceleration Feed Forward Factor</i>	Factor de aceleración previo del controlador de posición

Tabla A.1: Parámetros de calibración del controlador.

En el caso de realizarse una calibración en modo *Expert Tuning* en la cual se ajusten los parámetros manualmente el primer parámetro que se ha de ajustar es la constante P.

La variable de salida del sistema, ya sea de corriente de velocidad o de posición, se multiplica por la ganancia de proporcionalidad P y se introduce en la realimentación del sistema. Si el valor de P es demasiado elevado el lazo se vuelve inestable, por lo tanto, con la constante P nunca se podrá en condiciones estáticas (de no variaciones en la demanda) alcanzar el valor deseado sino uno algo inferior.

Para compensar la falta de ganancia en la regulación se añade a la señal que excita los estatores un *offset* cuyo valor no es constante en el tiempo.

La señal de retorno de lazo es analizada por un circuito integrador. La salida de dicho circuito es multiplicada por la constante de integración I obteniéndose de esta manera el valor *offset* que se sumará a la señal del circuito de proporcionalidad.

En el controlador de posición además se puede configurar la constante D. Cuando hay variaciones bruscas en la posición demandada se produce una sobrereacción por parte del controlador PI que hace que la posición real describa un movimiento oscilatorio en torno a la posición demandada.

Para evitar que se produzca este fenómeno se implementa un circuito derivador que compensa dicha sobrereacción y que logra que el controlador responda igual de rápidamente en alcanzar la nueva posición demandada pero amortiguando dicha oscilación inicial. La señal de salida del circuito derivador es multiplicada por una ganancia denominada parámetro D.

Una vez realizado el proceso de calibración los parámetros calculados son almacenados en el controlador de forma permanente.

A.4. Control del motor de rotación mediante API.

A.4.1. Introducción.

Maxon motor facilita una API (*Application Programming Interface*) disponible en su página web (<http://www.maxonmotor.es>). Para descargar el software hay que seleccionar el controlador EPOS2 70/10 y posteriormente en la sección de descargas seleccionar *EPOS Windows 32-Bit and 64-Bit DLL*.

En el paquete software se incluyen las librerías necesarias para distintos lenguajes de programación: Microsoft Visual C++, Microsoft Visual Basic, Borland C++, Borland Delphi y LabWindowsCVI. En este documento se hará referencia exclusivamente a la librería de Microsoft Visual C++.

A.4.2. Instalación de la librería.

Lo primero que hay que hacer es configurar el entorno de programación para que tanto el compilador como el vinculador sean capaces de gestionar la librería. Para ello se ha de acceder a las propiedades del proyecto y en la sección *C/C++* y apartado *Directorio de inclusión adicionales* se ha de introducir la ruta de acceso donde se ubica el archivo de cabecera *Definitions.h*.

Posteriormente, en la sección *Vinculador* y apartado *Dependencias Adicionales* se ha de introducir la ruta de acceso donde se ubica el archivo *EposCmd.lib* o el archivo *EposCmd64.lib* (en función de la arquitectura utilizada: 32 ó 64 bits). Además se ha de copiar en la carpeta donde reside el proyecto el archivo *EposCmd.dll* (*EposCmd64.dll*

para la versión de 64 bits). En el código del proyecto se ha de añadir la linea `#include "Definitions.h";`

Una vez configurado el entorno de programación se podrán realizar llamadas a las distintas funciones disponibles en la librería.

A.4.3. Comunicación con el controlador EPOS2.

La librería permite comunicarse con el controlador EPOS2 mediante cualquiera de los interfaces de los que dispone el mismo: RS232, CANopen y USB (véase 3.2.2.3). En los siguientes apartados se hará referencia exclusivamente a la comunicación USB.

El controlador dispone de un banco de microinterruptores que permiten configurar el número de nodo asociado. Dicho número de nodo es utilizado para el correcto direccionamiento de los mensajes entre la librería y los diferentes controladores que puedan coexistir en el mismo bus.

Para establecer comunicación con el controlador vía USB se ha de utilizar las siguiente linea de código:

```
motorHandle = VCS_OpenDevice("EPOS2","MAXON SERIAL V2","USB","USB0",&ErrorCodeMaxon);
```

La variable de retorno *motorHandle* es del tipo *HANDLE* y se utilizará en los diferentes comandos de la librería para referirse a dicha conexión.

La variable de retorno *ErrorCodeMaxon* es del tipo *DWORD* y es utilizada para la librería para indicar los posibles errores en el intento de conexión. Un valor nulo indica que la conexión se ha realizado con éxito.

A.4.4. Selección del modo de operación.

Mediante la siguiente linea de código se le indica al controlador el modo de operación (véase la sección A.2):

```
VCS_SetOperationMode(motorHandle, node, operationMode, &ErrorCodeMaxon);
```

La variable *node* es de tipo *WORD* e identifica el número de nodo del controlador. La variable *operationMode* es de tipo *int8* y establece el modo de operación. En la tabla A.2 se muestran los distintos códigos asociados a los distintos modos de operación.

A.4.5. Excitación de los bobinados.

Para que el controlador responda correctamente a las posibles peticiones de giro a una velocidad determinada, de búsqueda de la posición de inicio (*Homing*) o de movimiento hasta una determinada posición los bobinados del motor deben estar excitados (estado del controlador *Enable*).

VALOR	MODO DE OPERACIÓN
0x06	Homing Mode
0x03	Profile Velocity Mode
0x01	Profile Position Mode
0xFF	Position Mode
0xFE	Velocity Mode
0xFD	Current Mode
0xFA	Step/Direction Mode

Tabla A.2: Modos de operación del controlador EPOS2 70/10.

Si el controlador está en modo *Enable* y se encuentra en un modo de operación en el cual tiene que mantener una posición determinada(Position Modo o Homing) tratará de mantener dicha posición ofreciendo resistencia si se trata de mover el eje del rotor.

Mediante la función:

```
VCS_SetEnableState(motorHandle, node, &ErrorCodeMaxon);
```

Se activa el estado *Enable*. Para desactivar la excitación de los bobinados de ha de llamar a la función:

```
VCS_SetDisableState(motorHandle, node, &ErrorCodeMaxon);
```

A.4.6. Inicio de la rotación.

Cuando el controlador está *Enable* y en modo de operación *Profile Velocity Mode* se puede iniciar la rotación. Para ello previamente se ha de configurar la aceleración y deceleración del movimiento mediante la función:

```
VCS_SetVelocityProfile(motorHandle, node, acceleration, deceleration, &ErrorCodeMaxon);
```

Tanto *acceleration* como *deceleration* son dos variables de tipo DWORD y sus unidades son $\frac{rpm}{s^2}$. Mediante la función:

```
VCS_MoveWithVelocity(motorHandle, node, velocity, &ErrorCodeMaxon);
```

Se inicia el movimiento del motor. La variable *velocity* es de tipo long y sus unidades son rpm. Para detener la rotación hay que llamar a la función:

```
VCS_HaltVelocityMovement (motorHandle, node, &ErrorCodeMaxon);
```


Apéndice B

Documentación tarjeta de síntesis de señales de reloj.

B.1. Módulo MOD5270.

Tanto la tarjeta controladora de la fuente de alimentación diseñada (apartado 2.3) como la tarjeta de síntesis (apartado 5.1.1.5) requieren de un microcontrolador para realizar diversas tareas tales como generar señales de disparo, configurar diferentes dispositivos y comunicarse con la aplicación del sistema (véase el apartado 5.2).

El microcontrolador elegido para realizar dichas tareas es de la marca Freescale y modelo Coldfire 5270. Tiene una arquitectura *RISK* (*Reduced Instruction Set Computing*) de 32 bits. Soporta una frecuencia de reloj máxima de 150 MHz y tiene una memoria SRAM de 64 KB. En la figura B.1 se muestra un diagrama del mismo.

Incorpora un controlador *FEC* (*Fast Ethernet Controller*), un controlador *DMA* (*Direct-Memory Access*), 4 temporizadores programables de 32 bits, un controlador *QSPI* (*Queued Serial Peripheral Interface*), un controlador I2C (*Inter-Integrated Circuit*) y un controlador UART (*Universal Asynchronous Receiver-Transmitter*). Permite seleccionar como fuente de reloj un cristal de cuarzo o una bien una señal externa. Dicha selección se realiza estableciendo unos niveles lógicos en determinados pines del encapsulado.

El microcontrolador se integra dentro de un módulo del fabricante Netburner y modelo MOD5270 (figura B.2). El módulo integra el micro conectado a un cristal de 147.5 MHz, una memoria de 8 MB *SDRAM* (*Synchronous Dynamic Random-Access Memory*) y una memoria de 512KB *FLASH*. El módulo se alimenta mediante una tensión de 3.3 V.

Netburner facilita además del hardware un sistema operativo *multithread* denominado *μuC/OS* desarrollado en *C*, una librería de funciones *C* denominada *RTOS* (*RunTime*

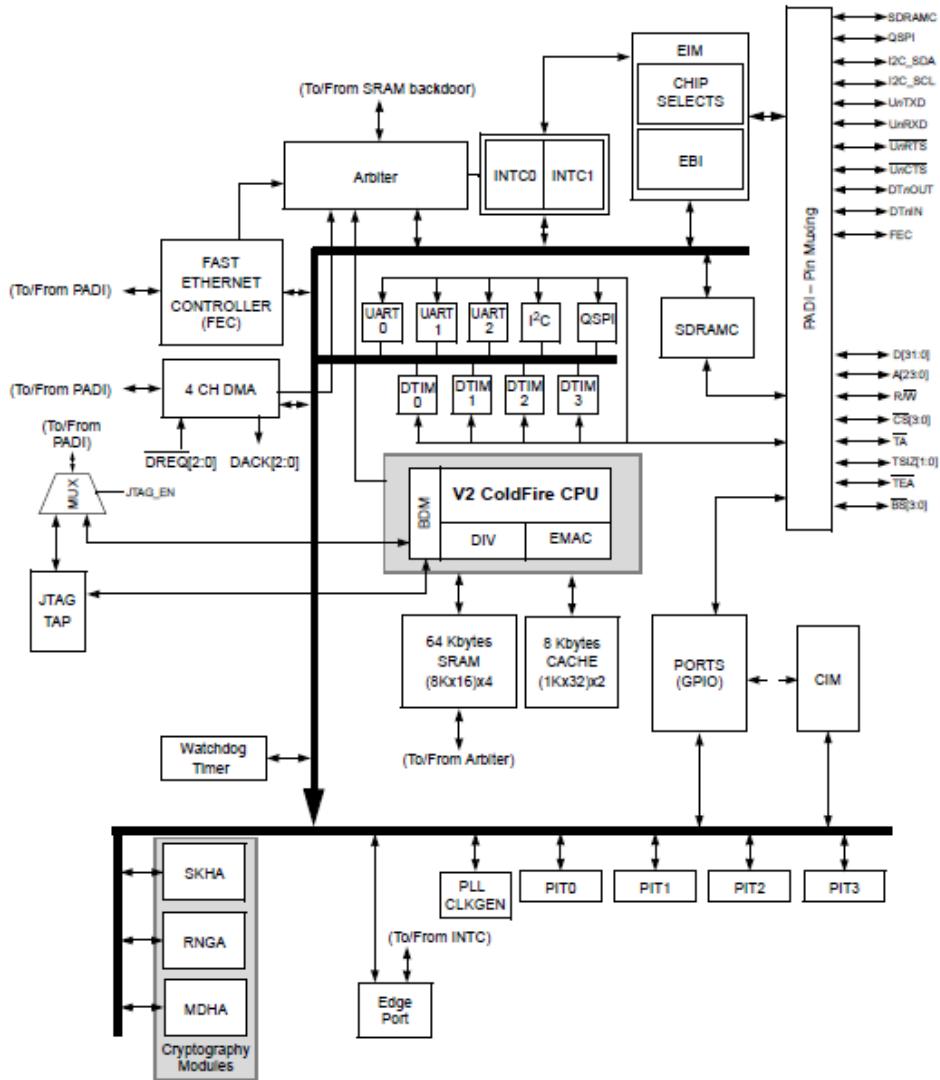


Figura B.1: Arquitectura del microcontrolador MC5270.

Operating System) que simplifica el manejo del microcontrolador y un compilador C. La librería RTOS permiten realizar las siguientes funciones:

- Protocolo IPV4 (Internet Protocol).
- Servidor/Cliente DHCP (Dynamic Host Configuration Protocol).
- Protocolo UDP (User Datagram Protocol).
- Protocolo TCP (Transmission Control Protocol).
- Protocolo PPP (Point-to-Point Protocol).



Figura B.2: Módulo 5270.

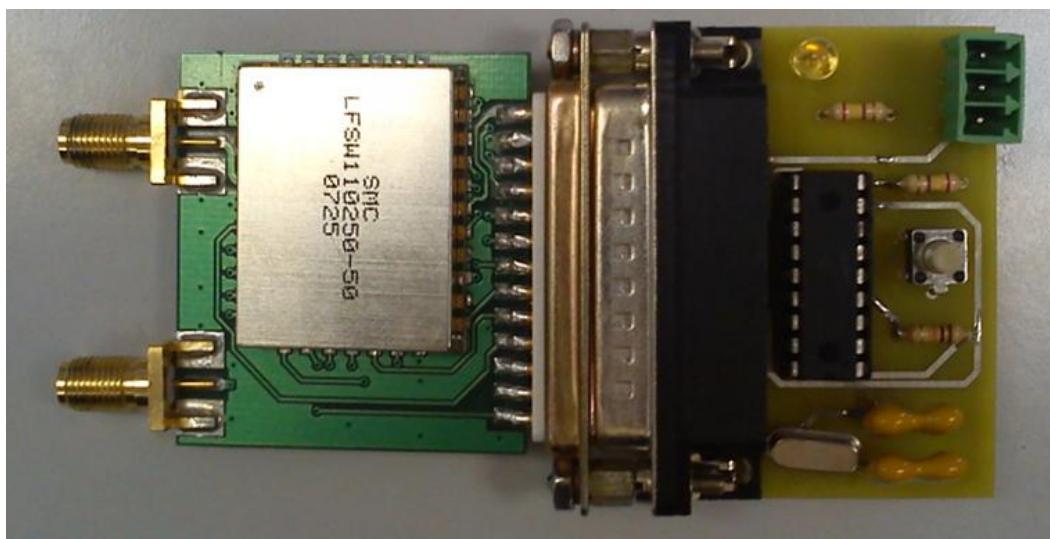
- Servidor/Cliente FTP (File Transfer Protocol).
- Servidor web HTTP (Hypertext Transfer Protocol).
- Servidor SMTP (Simple Mail Transfer Protocol).
- Cliente POP3 (Post Office Protocol 3).
- Controlador de bus I2C (Inter-Integrated Circuit).
- Controlador UART(Universal Asynchronous Receiver-Transmitter).
- Controlador de bus CAN (Controller Area Network).
- Protocolo SNMP (Simple Network Management Protocol).
- Protocolo SSL (Secure Sockets Layer).
- Protocolo SSH (Secure SHell).
- Gestor de *threads*.

La tarjeta MOD5270 no deja accesibles al usuario los pines que permiten seleccionar el tipo de reloj (Cristal o fuente externa) ya que éstos quedan ocultos entre la PCB y el encapsulado del microcontrolador. Por este motivo el microcontrolador solo puede trabajar con el cristal que incorpora. Además, en el caso de que se desmontara el chip y modificara la tarjeta para poder usar una fuente de reloj externa el software dejaría de ser operativo en el caso de que la nueva señal de reloj tenga una frecuencia distinta a la original.

Esta limitación se debe a que funciones del sistema operativo, como la gestión del protocolo IP, incorporan temporizaciones que están escritas explícitamente para dicha frecuencia de reloj.

B.2. Programación del sintetizador mediante PIC.

CONFIGURACIÓN DEL SINTETIZADOR LFSW110250-50 MEDIANTE SPI CON PIC16F84A



Grupo de Microondas y Radar.
Alejandro Badolato.
Marzo 2011.

Contenido

1.	INTRODUCCIÓN.....	3
2.	DISEÑO DE LA PCB Y PRINCIPIO DE FUNCIONAMIENTO.....	4
3.	DESCRIPCIÓN DE LAS SEÑALES	5
4.	DIAGRAMAS DE FLUJO.....	7
5.	INSTALACIÓN DEL ENTORNO	9
6.	GRABACIÓN DEL MICROCONTROLADOR	12
7.	CÓDIGO DEL PROGRAMA.....	14
8.	ESQUEMAS ELÉCTRICOS.....	17
9.	POSIBLES MEJORAS	19

1. INTRODUCCIÓN

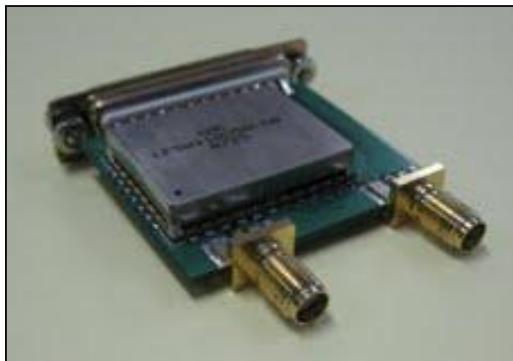
Con este documento se pretende describir la forma de programar el sintetizador LFSW1050 de Synergy mediante interfaz SPI (Serial Peripheral Interface).

El sintetizador dispone de una placa de evaluación dotada de un conector DB25 por el que recibe alimentación e interconexiona todas sus entradas y salidas a excepción de la señal de referencia y la de salida de RF, ambas conectorizadas con SMA.

El fabricante facilita una aplicación dotada de GUI que permite programar el dispositivo mediante SPI utilizando un puerto paralelo, una interfaz obsoleta en los equipos actuales. Además, en equipos portátiles la aplicación no funciona correctamente por tener limitados en corriente los pines del puerto paralelo.

A pesar de que en la documentación técnica del sintetizador figuran cadenas de programación que permiten una configuración no volátil mediante In-Circuit programming, el fabricante no suministra la opción. En consecuencia, es necesario instalar un equipo de sobremesa junto al sintetizador e introducir la cadena de configuración en cada puesta en marcha del sistema.

Todos estos inconvenientes han provocado que se haya tomado la decisión de prescindir de la herramienta del fabricante y buscar una solución que permita una programación automática en cada reset del equipo. Para ello se ha desarrollado una placa de circuito impreso (Printed Circuit Board, en adelante PCB) que integra un microcontrolador PIC16F84A, encargado de estas labores.



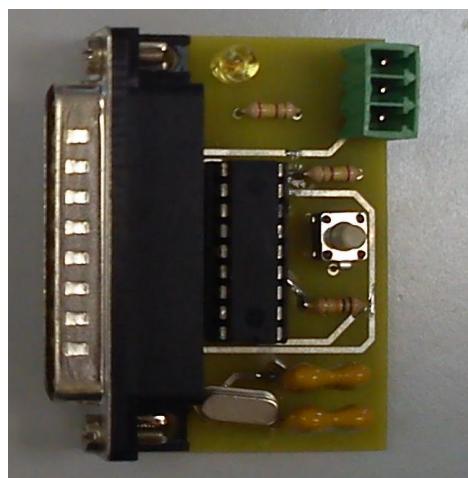
[1] Placa de evaluación del LFSW1050.



[2] Graphic User Interface para la programación del LFSW1050 utilizando puerto paralelo.

2. DISEÑO DE LA PCB Y PRINCIPIO DE FUNCIONAMIENTO

La PCB [3] requiere una alimentación cuyo nivel de tensión es +5Vcc. Dicha alimentación la recibe mediante un conector enchufable con la que alimenta tanto al PIC como al sintetizador. Dispone de un pulsador para poder realizar un reset al sintetizador que es vigilado por el PIC mediante espera activa. Una vez detectado el reset por software, se envía una señal de reset al equipo. Tras un retardo establecido se le envía mediante SPI la cadena de configuración prefijada que sale de la tarjeta a través del conector DB25-M.



[3] Programador del sintetizador.

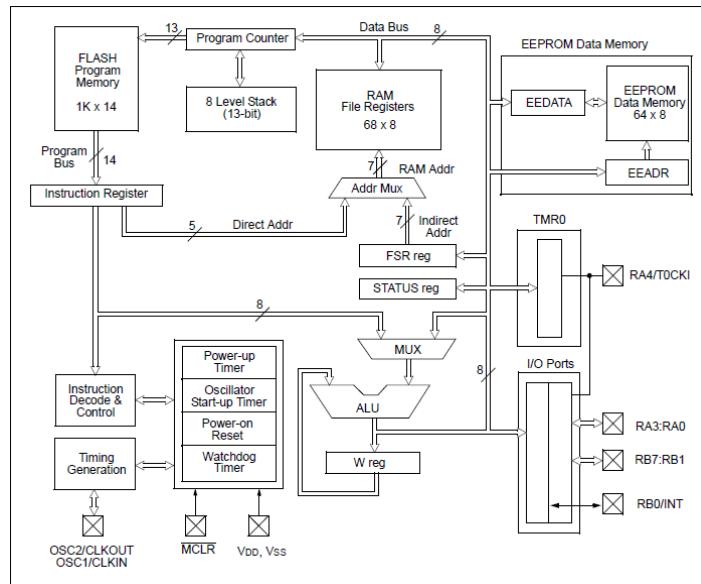
Además de la funcionalidad anteriormente descrita, cuando el PIC recibe alimentación, tras un retardo de 800 ms envía al sintetizador la cadena prefijada por lo que no es necesaria la intervención del usuario.

El sintetizador facilita una salida CMOS por la que indica si el sintetizador está enganchado a la frecuencia establecida. Se ha instalado un led en la PCB que se enciende cuando el sintetizador está enganchado.

El sintetizador tiene un consumo aproximado de 120mA. El consumo del PIC es despreciable y el del LED en estado activo es de 25 mA. Puesto que el consumo total es muy bajo, el ancho de las pistas elegido en el diseño de la PCB es de tan sólo 0.8 mm, permitiendo un tamaño de PCB más pequeño.

3. DESCRIPCIÓN DE LAS SEÑALES

Las señales que dotan de funcionalidad al sistema son los pines del PIC RB0, RB1, RB2, RB3 y RB4, todos ellos del puerto B. El único que está programado como entrada y dotado de pull-up interno es RB3, que se encarga de detectar el pulsador de la PCB.



[4] Diagrama de bloques del PIC16F84A

En la figura [17] pueden localizarse las señales y su interconexión entre el conector DB25 y el PIC. A continuación, se incluye una breve descripción de las señales en cuestión:

RESETIN: Mediante pull-up interno, la señal de entrada al PIC está normalmente a nivel alto. Cuando se presiona el pulsador integrado en la PCB (S1; ver figura [1]) baja el nivel a 0V, provocando que la condición analizada por la espera activa en el código del PIC se cumpla y por tanto se envíe la señal *RESETOUT* al sintetizador.

RESETOUT: Señal de reset generada por el PIC que se envía al sintetizador. El sintetizador tiene pull-up que mantiene normalmente la señal a nivel alto. Cuando se detecta nivel bajo en *RESETIN* el PIC pone a nivel bajo la señal *RESETOUT* indicándole al sintetizador la orden de reset.

CLOCK (SCK): Reloj generado por el PIC para la programación SPI. Los flancos de subida de dicho reloj validan la recepción de un bit por el pin de datos en el sintetizador, siempre y cuando la señal *LATCHENABLE* se encuentre en nivel bajo. Para no introducir ruido al sintetizador, la señal de reloj se activa sólo cuando el PIC tiene que enviar una cadena de

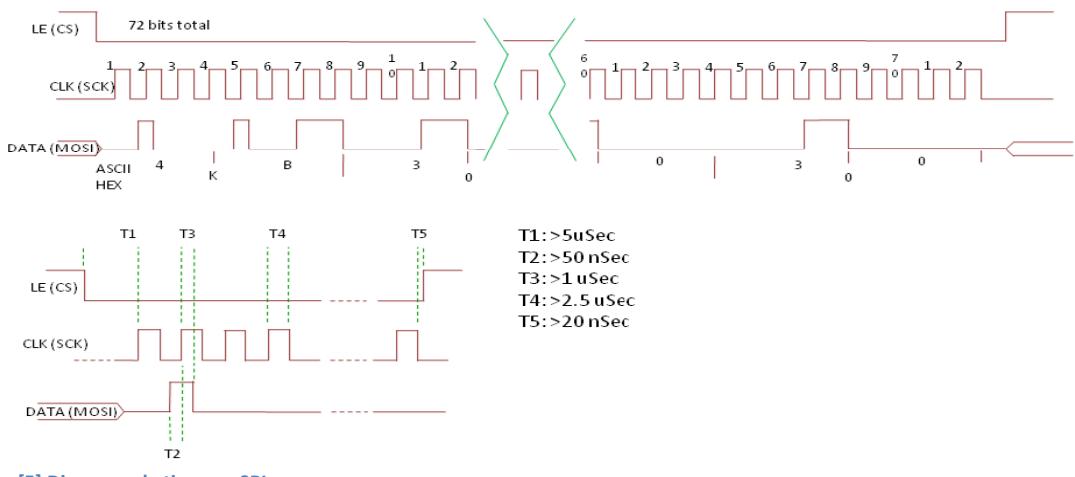
datos. En estado de reposo (cuando no se está enviando ninguna cadena de configuración al sintetizador), la señal se mantiene en nivel alto. El tiempo de bit del reloj es de 2 ms.

DATA (MOSI): Salida de datos del PIC hacia el sintetizador. Se transmiten los bits con codificación Little-Endian, y dentro de cada nibble (medio octeto), se transmite primero el bit más significativo. En primer lugar se transmiten los bits correspondientes al primer carácter por la izquierda de la cadena de configuración del sintetizador almacenada en el PIC.

LATCHENABLE (CS): Normalmente se encuentra en nivel alto por orden del PIC. Es la señal de validación de los datos. Cuando el PIC fija CS a nivel bajo, el sintetizador capturará cada bit del canal *DATA* en los flancos de subida del reloj *SCK*.

ERROR FLAG: El sintetizador indica a través de esta salida la ocurrencia de un error en la lectura de la cadena recibida a través del canal *MOSI*. Dispone de un pull-up interno en el sintetizador de $20\text{k}\Omega$. Además, siguiendo instrucciones del fabricante, se le ha instalado una resistencia pull-down de 140Ω en el circuito impreso que alberga el PIC para poder analizar la señal mediante un osciloscopio. Según se van recibiendo nibbles, el sintetizador analiza si la cadena recibida es coherente con alguna de las posibles configuraciones permitidas. Si detecta algún error, al finalizar el nibble en cuestión, mantiene a nivel alto la señal durante un ciclo de reloj *SCK*. Esta señal no es utilizada por el PIC.

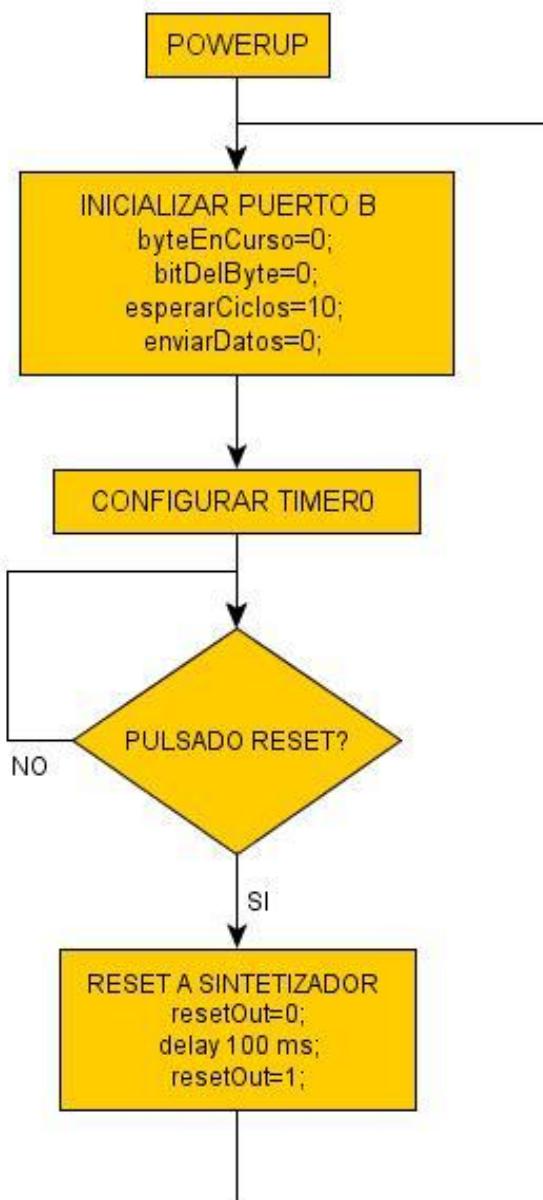
En la siguiente figura se muestra el diagrama de tiempos establecido por el sintetizador:



[5] Diagrama de tiempos SPI.

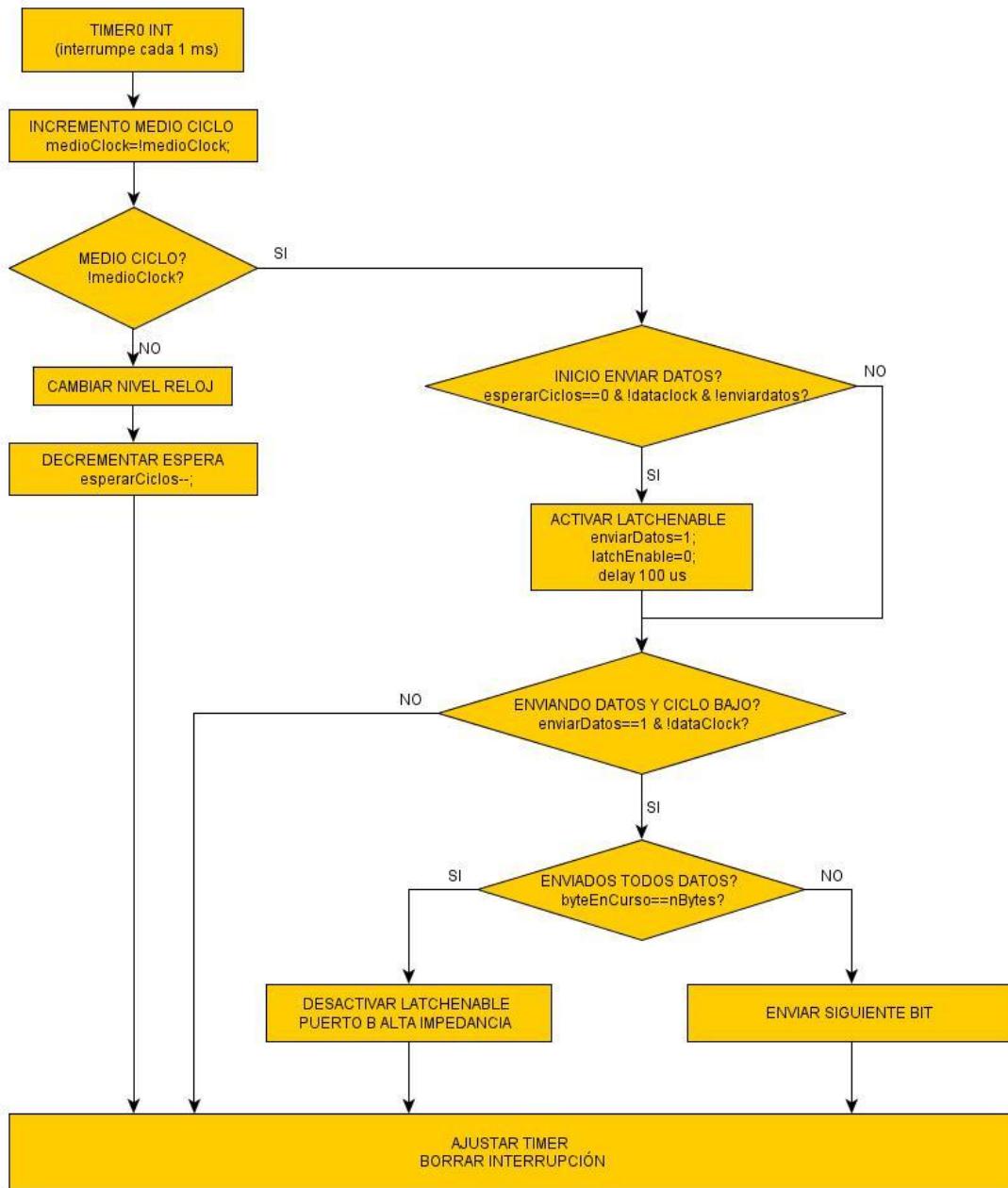
4. DIAGRAMAS DE FLUJO

A continuación se presentan los diagramas de flujo que describen el comportamiento funcional del programa que se ejecuta en el PIC. Tras la inicialización del microcontrolador, se ejecuta el bloque POWERUP.



[6] Diagrama de flujo del programa principal.

A continuación se muestra la rutina de interrupción asociada al overflow del TIMERO.

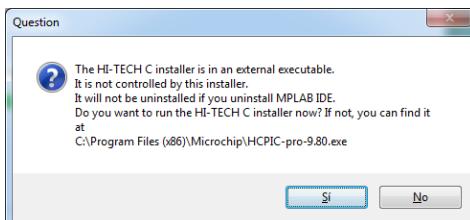


[7] Diagrama de flujo de la rutina de atención del timer.

5. INSTALACIÓN DEL ENTORNO

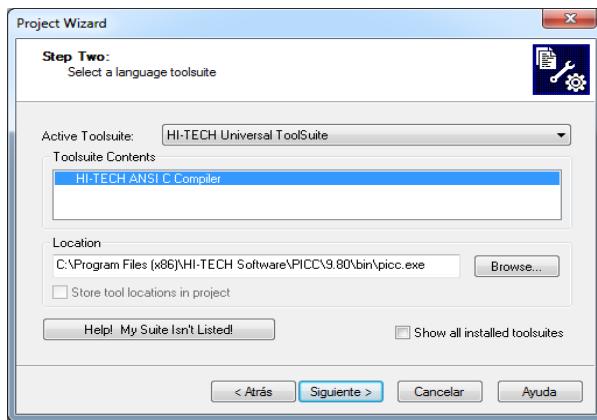
El entorno de trabajo para programar el microcontrolador es el que ofrece gratuitamente el fabricante del mismo. Se denomina MPLAB y se puede obtener en www.microchip.com

La versión utilizada para este proyecto ha sido MPLAB v8.63. Durante la instalación, el asistente ofrece instalar también HI-TECH (Compilador de C para PIC) [9]. Es necesario instalar dicho software para que el entorno pueda compilar el código de la familia de microcontroladores PIC.



[8] Instalación del compilador.

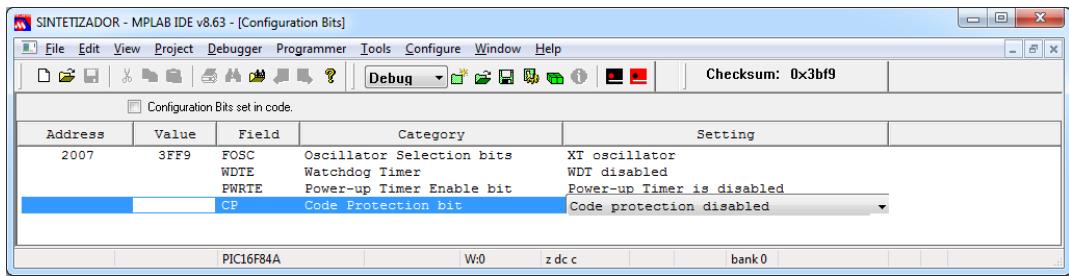
Tras la instalación del entorno y del compilador, hay que configurar el entorno correctamente. Para ello, es necesario ejecutar el asistente *Project Wizard* [9] ubicado en el menú *Project*. Se ha de seleccionar el dispositivo PIC16F84A. Posteriormente, se ha de pulsar en el botón *Siguiente* al observar la pantalla que se muestra a continuación.



[9] Configuración del entorno.

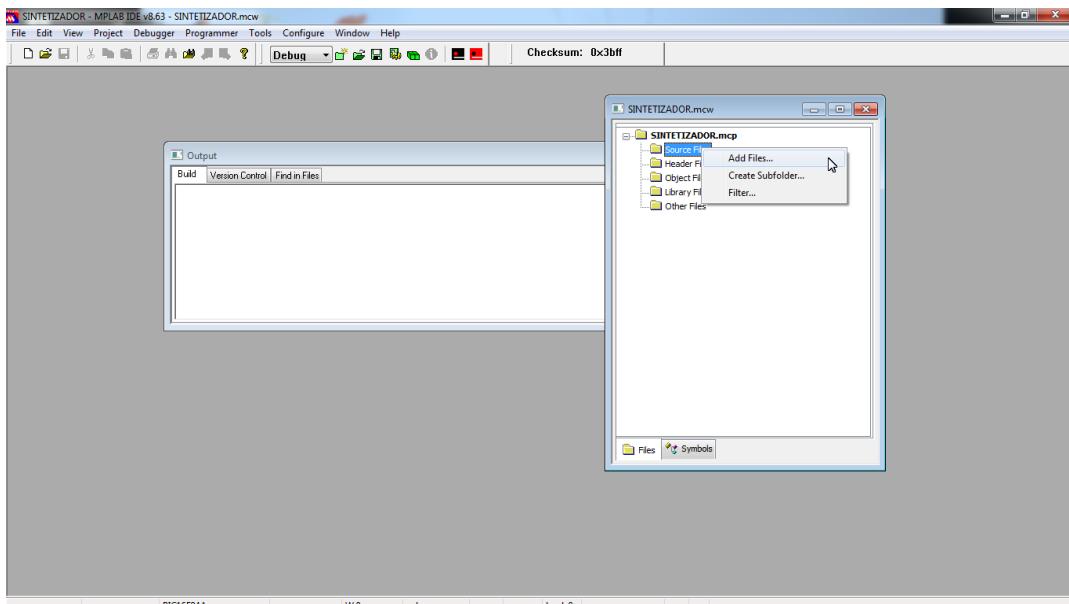
En la siguiente pantalla se debe poner un nombre al proyecto y pulsar repetidamente *Siguiente* hasta finalizar el asistente.

En el menú *Configure*, se ha de pulsar en *Configuration Bits*, quitar la opción *Configuration Bits set in code* y seleccionar las opciones marcadas en la pantalla:



[10] Configuration bits.

En la ventana donde se gestionan los archivos del *Workspace*, se debe pulsar sobre *Source Files* y seleccionar *Add Files*.



[11] Archivos del workspace.

Una vez creado el archivo, se puede escribir el código del programa. Si todo está correcto, el compilador debe mostrar el texto correctamente formateado [12]. Si no ocurre esto, es porque el archivo a editar se encuentra en *Other Files* en lugar de *Source Files*.

```

#include <htc.h> // Necesario para rutinas de retardo
#define TMR0_PRESCALER 0 //prescaler 1:2, ud timer=1x2=2 us
#define POLLING_PERIOD 200 //con reloj 4Mhz, 2us x 100 = 200 us interrumpimos
#define TMR0_SETTING 0xff - POLLING_PERIOD
#define XTAL_FREQ 4000000
#define DATALOCK R0
#define DATA_R0
#define LATCHENABLE RB2
#define RESETIN RB3
#define RESETOUT RB4
#define RETARDO_RESET 100 //mseg
#define RETARDO_INICIAL 800 //mseg
#define RESET=0 //definimos reset a nivel bajo
#define NO_RESET=1

/*
    PIN_PIC      PIN_DB25   SEÑAL
    RB0          5          DATA_CLOCK
    RB1          6          DATA
    RB2          8          LATCH_ENABLED
    RB3          9          RESET_IN
    RB4          9          RESET_OUT (entrada sintetizador)
    RB4 10       2,15,17,22 GND
    RB4 11       3,4,16     Vcc
    RB4 12       10         LOCK_DETECT
*/

***** ENVIO DE DATOS MEDIANTE SPI CON PIC16F84 PARA
***** SINTETIZADOR LFS SYNERGY
| ALEJANDRO BADOLATO           GMR 23/03/2011
***** ****
*/

```

[12] Código de programa formateado.

Para compilar, se necesita pulsar en el icono sobre el que se encuentra el ratón (ver figura superior [12]).

```

SINTETIZADOR - MPLAB IDE v8.63 - [Output]
Build | Version Control | Find in Files | Debug | Checksum: 0x3923
Build C:\Users\Aco\Documents\ETSIT\GMR\PICT\SINTETIZADOR for device 16F84A
Using driver C:\Program Files (x86)\HI-TECH Software\picc\3.80\bin\piccc.exe

Make: The target "C:\Users\Aco\Documents\ETSIT\GMR\PICT\main.p1" is out of date.
Executing: "C:\Program Files (x86)\HI-TECH Software\picc\3.80\bin\piccc.exe" -pass1 C:\Users\Aco\Documents\ETSIT\GMR\PICT\main.c -q -chip=16F84A -P -runtime=default -opt=debug -D__DEBUG=1 -g -esmlist * -errf
Executing: "C:\Program Files (x86)\HI-TECH Software\picc\3.80\bin\piccc.exe" -pass1 C:\Users\Aco\Documents\ETSIT\GMR\PICT\main.map -summary=default -output=default main.p1 -chip=16F84A -P -runtime=default -opt=debug -D_
(1273) Omniscient Code Generation not available in Lite mode (warning)
HI-TECH C Compiler for PIC10/12/16 MCUs (Lite Mode) V9.80
Copyright (C) 2010 Microchip Technology Inc.

Memory Summary:
Program space        used  100b (  384) of  400h words ( 37.5%)
Data space           used  2Ch (  44) of  44h bytes ( 64.7%)
EEPROM space         used  0h (  0) of  40h bytes (  0.0%)
Configuration bits   used  0h (  0) of  1h word (  0.0%)
ID Location space   used  0h (  0) of  4h bytes (  0.0%)

Running this compiler in PRO mode, with Omnidient Code Generation enabled.
produces code which is typically 40% smaller than in Lite mode.
The HI-TECH C PRO compiler (for this code) is 18h words smaller.
See http://microchip.hicsoft.com/portals/pic_pro for more information.

Loaded C:\Users\Aco\Documents\ETSIT\GMR\PICT\SINTETIZADOR.cof.

***** Build successful *****

PIC16F84A W:0 z dc c bank 0 WR

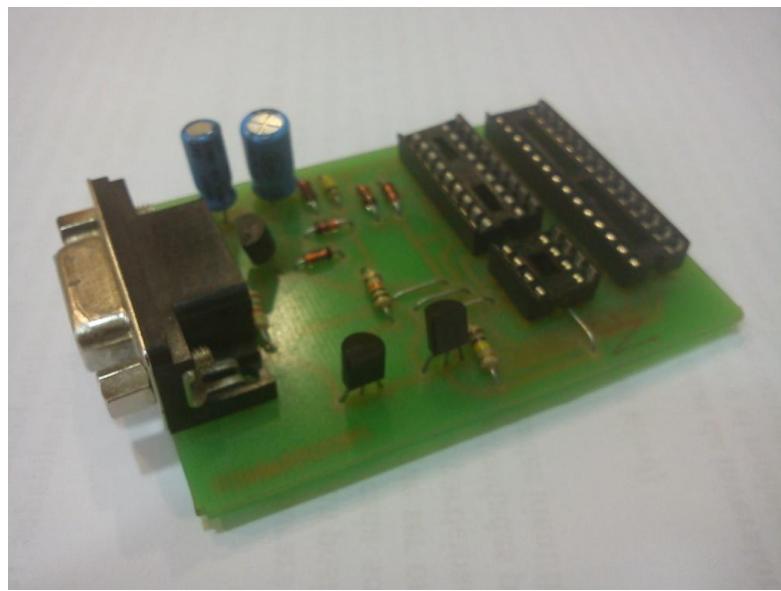
```

[13] Resultado del proceso de compilación.

En la carpeta del proyecto se genera un archivo .HEX con el mismo nombre del proyecto. Este es el archivo que se debe introducir en el grabador.

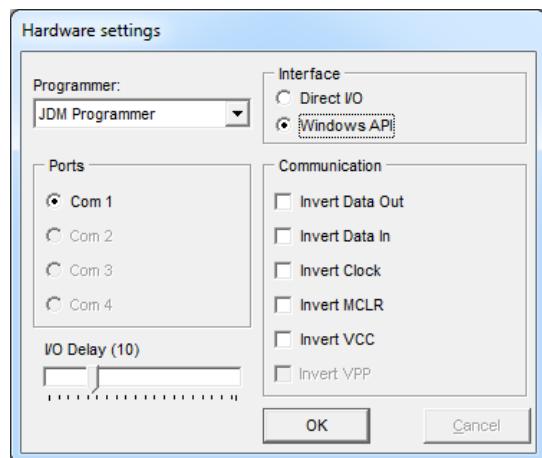
6. GRABACIÓN DEL MICROCONTROLADOR

El grabador del PIC [14] (disponible en el laboratorio C-406) utiliza un puerto COM para la programación del PIC. Para poder utilizar el grabador hay que descargar el software gratuito ubicado en <http://www.ic-prog.com>.



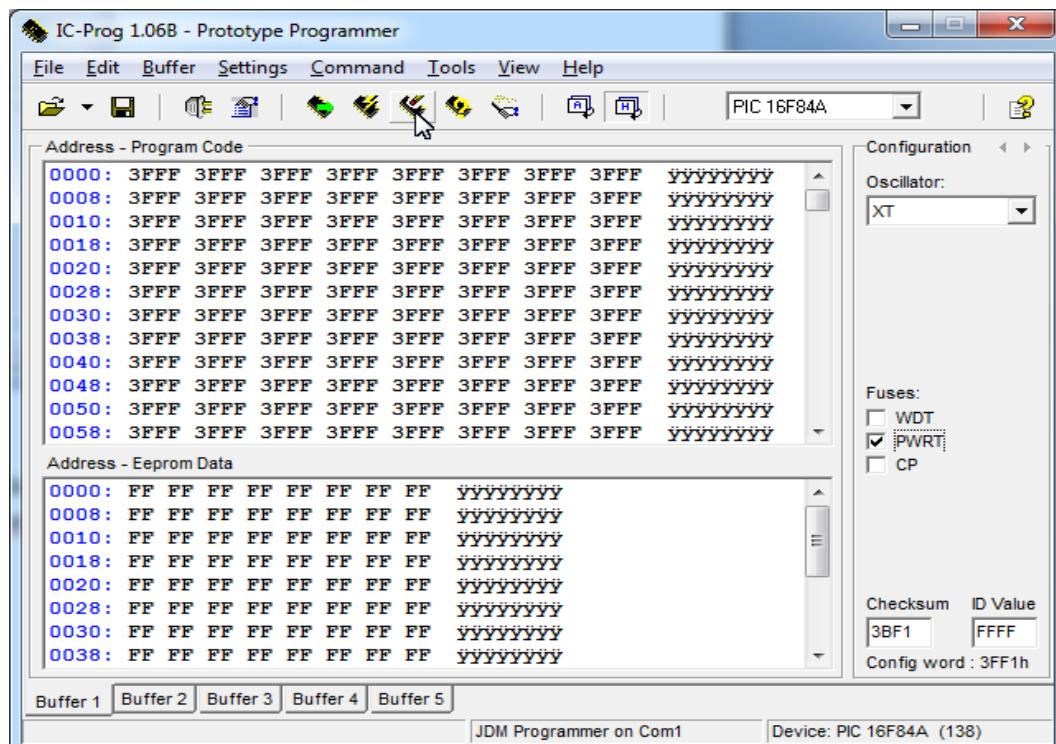
[14] Grabador del PIC.

La primera vez que se ejecuta el programa se requiere configurar el hardware con los parámetros reflejados en la siguiente figura [15]



[15] Configuración del hardware del grabador.

En el menú *File*, se debe seleccionar *Open File*, abrir el archivo .HEX, seleccionar en el menú desplegable el PIC16F84A. En el frame *Configuration* se ha de seleccionar *Oscillator XT* y *Fuses* según la pantalla mostrada en la figura siguiente [16].



[16]. Configuración del oscilador y fuses del PIC en el grabador.

Se debe pulsar en el ícono *Erase All* (indicado por la posición del ratón en la figura superior). El software nos indica que se ha borrado el dispositivo. A continuación, se tiene que pulsar en el botón *Program All*, el cual se identifica en la misma figura mediante el ícono de un rayo. El software graba el microcontrolador y, posteriormente, verifica el contenido.

7. CÓDIGO DEL PROGRAMA

```
include <htc.h> // Necesario para rutinas de retardo

#define TMRO_PRESCALER 0 //prescaler 1:2, ud timer=1x2=2 us
#define POLLING_PERIOD 200 //con reloj 4Mhz, 2us x 100 = 200 us interrumpimos
#define TMRO_SETTING (0xff - POLLING_PERIOD)
#define _XTAL_FREQ 4000000
#define DATACLOCK RB0
#define DATA RB1
#define LATCHENABLE RB2
#define RESETIN RB3
#define RESETOUT RB4
#define RETARDO_RESET 100 //mseg
#define RETARDO_INICIAL 800 //mseg
#define RESET=0 //definimos reset a nivel bajo
#define NO_RESET=1

/*
      PIN_PIC          PIN_DB25        SEÑAL
RB0    6              5             DATA_CLOCK
RB1    7              6             DATA
RB2    8              8             LATCH_ENABLED
RB3    9              XXX           RESET_IN
RB4    10             9             RESET_OUT (entrada sintetizador)
                           2,15,17,22   GND
                           3,4,16       Vcc
                           10            LOCK_DETECT

*****
***** ENVIO DE DATOS MEDIANTE SPI CON PIC16F84 PARA
***** SINTETIZADOR LFS SYNERGY
***** ALEJANDRO BADOLATO
***** GMR 23/03/2011
****/


volatile bit medioClock=0; // flag medio bit de reloj.
volatile int esperarCiclos=10; // Para generar 10 ciclos de reloj (DATACLOCK) antes de LE
volatile char bytes[19] = "4D3030303030393630"; //bytes de datos a enviar
volatile int nbytes = 18; // numero de bytes a tx.
volatile int byteEnCurso=0; //Puntero del proximo byte a transmitir
volatile int bitDelByte=0; //bit dentro del byte del proximo byte a trasmitir
volatile int dato; // variable local de enviadato. Se hace global para controlar en debug.
volatile bit enviarDatos=0; // flag que indica que se debe empezar a enviar datos
// declaramos funciones
void enviaDato (void);
```

```

void init(void);

void init(void){
    // puerto b todo salidas=0 menos reset_in=1
    TRISB=0x08; // rb3=RESET_IN como entrada resto salidas
    RBPU=0; // Activar pull ups portB
    LATCHENABLE=1; // Inactivo. Se activa poniendo un 0
    __delay_ms(RETARDO_INICIAL); // retardo inicial de 200 ms
    // delay ejecuta codigo. no usa TMRO ni WTDG
    // inicializamos variables tras reset
    medioClock=0;
    esperarCiclos=10;
    enviarDatos=0;
    byteEnCurso=0;
    bitDelByte=0;
    RESETOUT=NO_RESET; // anulamos reset
    // configuramos timer0
    // 0B11000000 = desactiva pullups y prescaler para timer0
    OPTION&=0B11000000; //desactivar ultimos 6 bits para congiurar TMRO
    OPTION|=TMRO_PRESCALER; //set prescaler 1:1
    TMRO=TMRO_SETTING;
    TOIE=1; // timer overflow interrupt
    GIE=1; // global interrupt enable
}

// Interrumpe cada cuarto de ciclo de reloj
// espera 10 ciclos de reloj antes de comenzar a enviar datos
// Al octavo ciclo activa LATCH ENABLE
// Llama a envia datos en mitad del ciclo bajo
// cuando se termina de enviar datos espera 2 ciclos y desactiva LE
void interrupt isr(void){
    if(TOIF){
        medioClock=!medioClock; // cambiamos cada IRQ
        if (!medioClock) {
            // fin de bit. Cambiar reloj
            DATA CLOCK=!DATA CLOCK;
            // decrementar espera
            if (esperarCiclos>0) {
                esperarCiclos--;
            }
        }
    } else {
        // mitad de ciclo alto o bajo
        if (esperarCiclos==0 && !DATA CLOCK && !enviarDatos){
            // enviamos datos
            enviarDatos=1;
            LATCHENABLE=0;
            __delay_us(100); // ver T1 en AN7100
            // si estamos en ciclo bajo en mitad bit y con LE enviamos dato
        }
        if (enviarDatos && !DATA CLOCK) enviaDato();
    }
    TMRO=TMRO_SETTING; // restablecemos contador timer para overflow
    TOIF=0; // borramos interrupcion
}
}

```

```

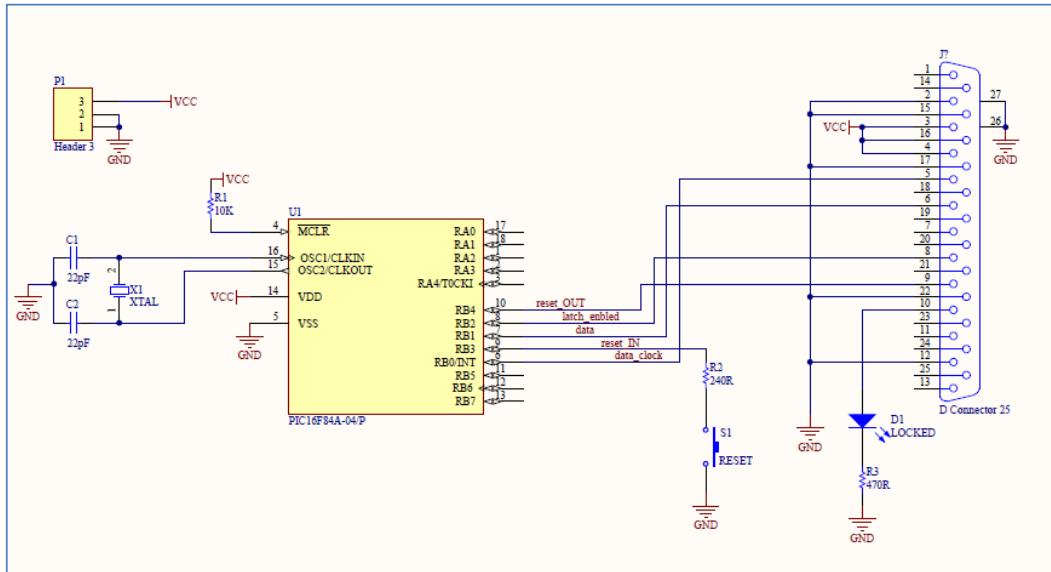
void main(void){
    init();
    while(1){
        // atendiendo interrupciones del reloj y vigilando reset
        if(RESETIN==0){
            GIE=0;           // turn off interrupts
            RESETOUT=0; // Activar reset a sintetizador
            __delay_ms(RETARDO_RESET);           // reset de 50 ms
            RESETOUT=1; // liberar reset
            init();
        }
    }
}

// envia un bit de datos. Si termina secuencia modifica esperarCiclos
// para que la IRQ desactive LE
void enviaDatos(void) {
    if (bitDelByte==4) {
        // cambiar de byte
        byteEnCurso++;
        bitDelByte=0;
    }
    if (byteEnCurso==nbytes){
        // se termino de enviar datos
        LATCHENABLE=1;
        TRISB=0xFF; // dejamos todos los pines en alta impedancia
        RBPU=0; // Activar pull ups portB
    } else{
        // enviar datos
        // en SPI se transmiten los bytes por orden y los bits de mas peso a menos
        dato=bytes[byteEnCurso];
        if (dato>57){
            // es letra A-F A=65-55=10 (VER ASCII)
            dato-=55;
        } else {
            // es numero 0-9 0=48-48=0 (VER ASCII)
            dato-=48;
        }
        if (bitDelByte==0) dato=dato&0x8;
        if (bitDelByte==1) dato=dato&0x4;
        if (bitDelByte==2) dato=dato&0x2;
        if (bitDelByte==3) dato=dato&0x1;
        if (dato>0) dato=1;
        DATA=dato;
        bitDelByte++;
    }
}

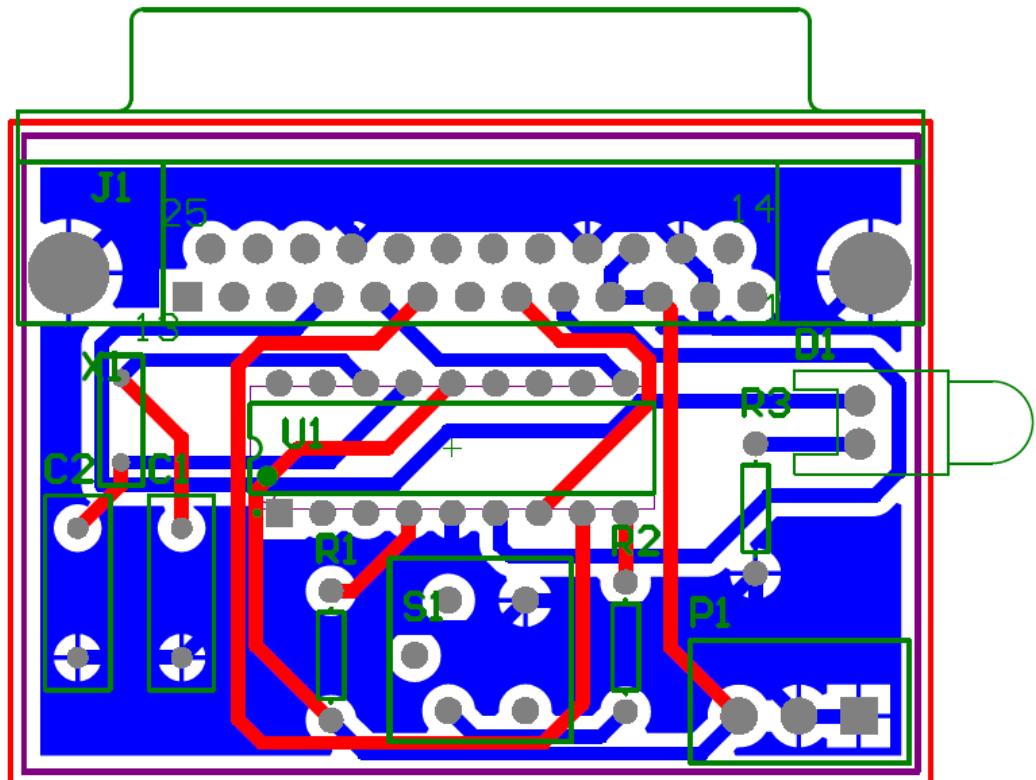
```

8. ESQUEMAS ELÉCTRICOS

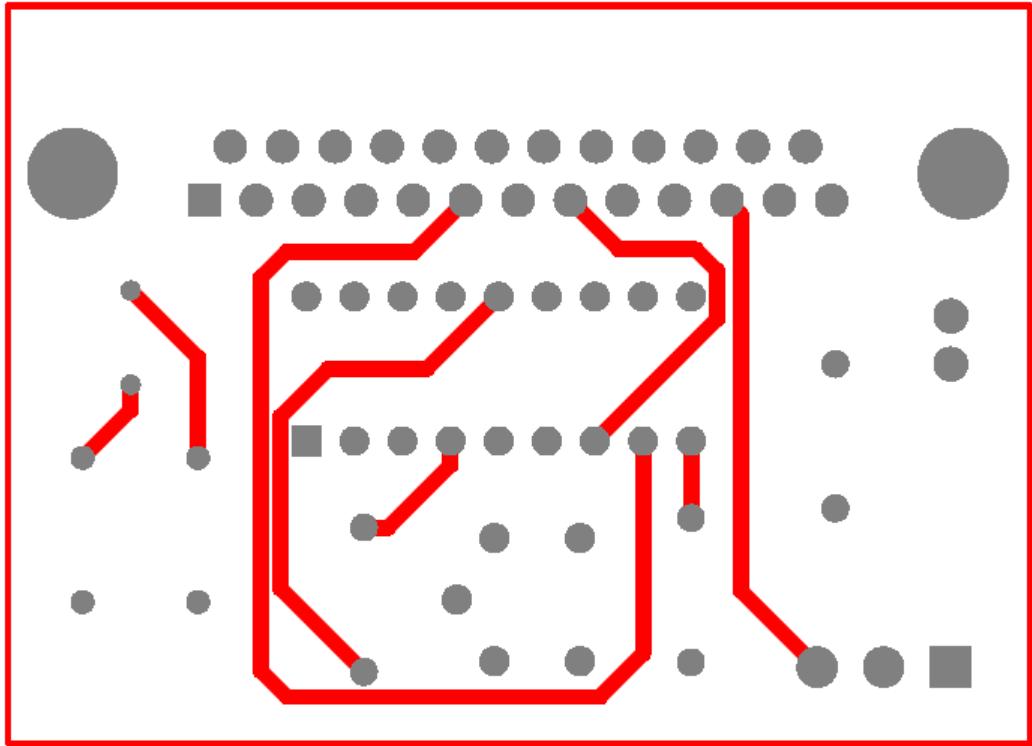
Para la elaboración de la PCB se ha utilizado como herramienta CAD la versión Summer09 del software Altium Designer. A continuación se muestran los diseños realizados.



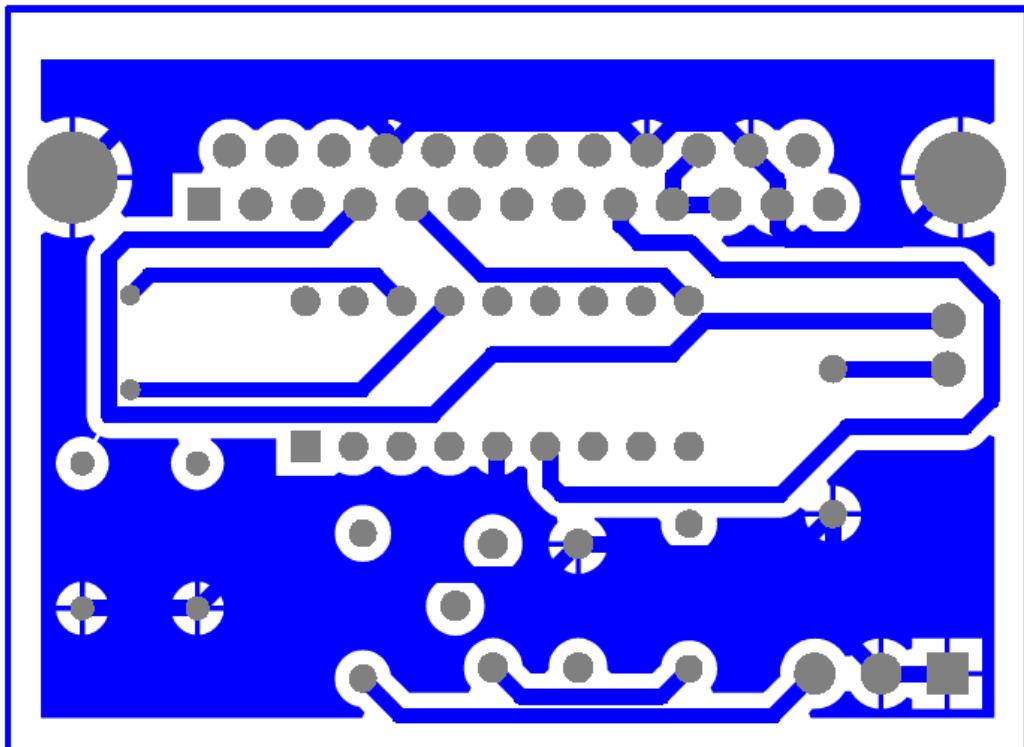
[17]. Esquema eléctrico de la PCB.



[18]. Layout de la PCB.



[19]. Layout de la top layer.



[20]. Layout de la bottom layer.

9. POSIBLES MEJORAS

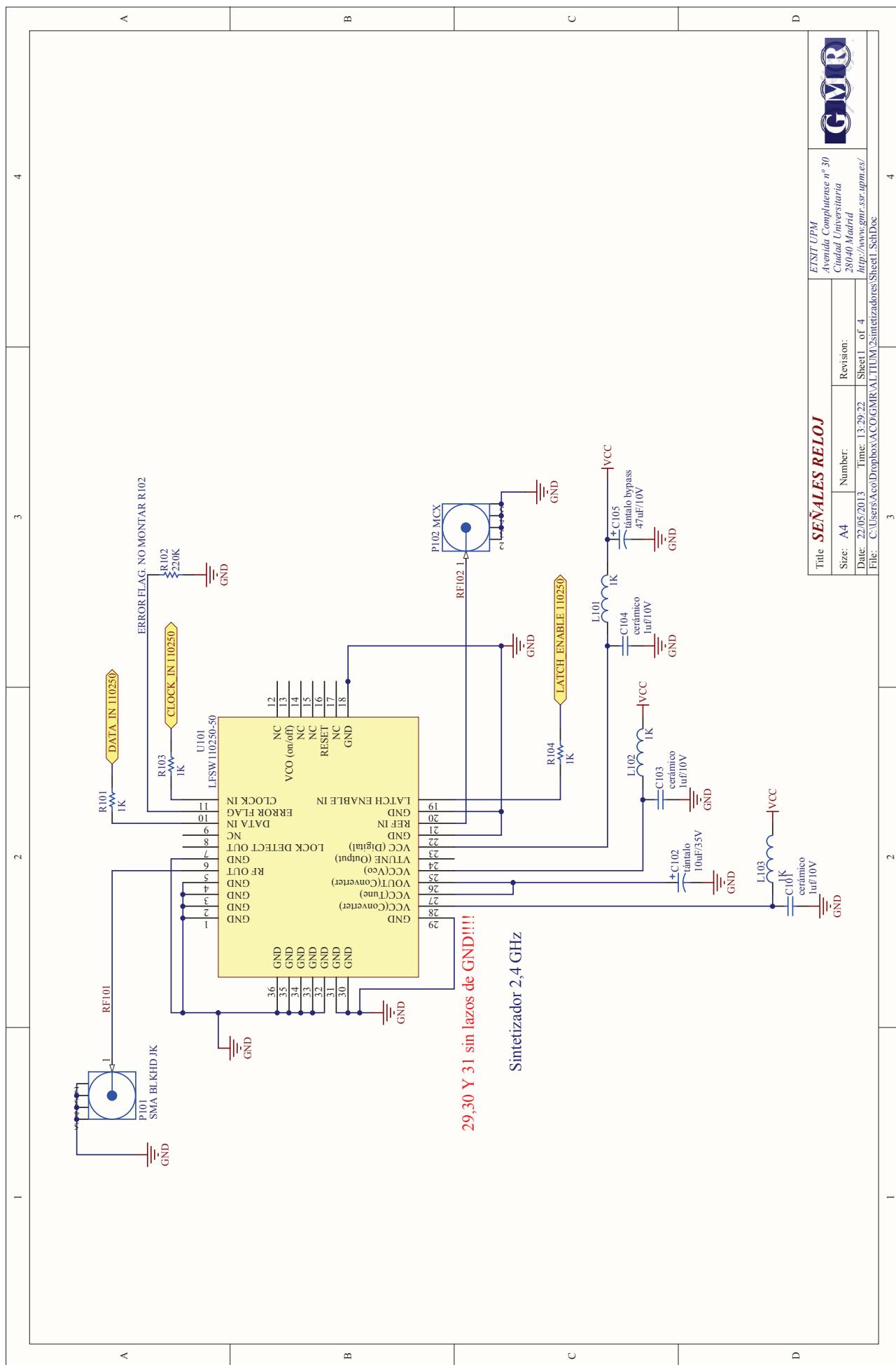
Una necesidad futura puede ser la de requerir enviar diferentes cadenas de inicialización al sintetizador con el fin de trabajar con distintos escenarios de operación. Dado que esta opción no se ha barajado inicialmente y, por tanto, no se ha dotado a la PCB de elementos de selección como pueden ser los microinterruptores dip switch, se deberá recurrir a estrategias más complejas para poder permitir dicha selección.

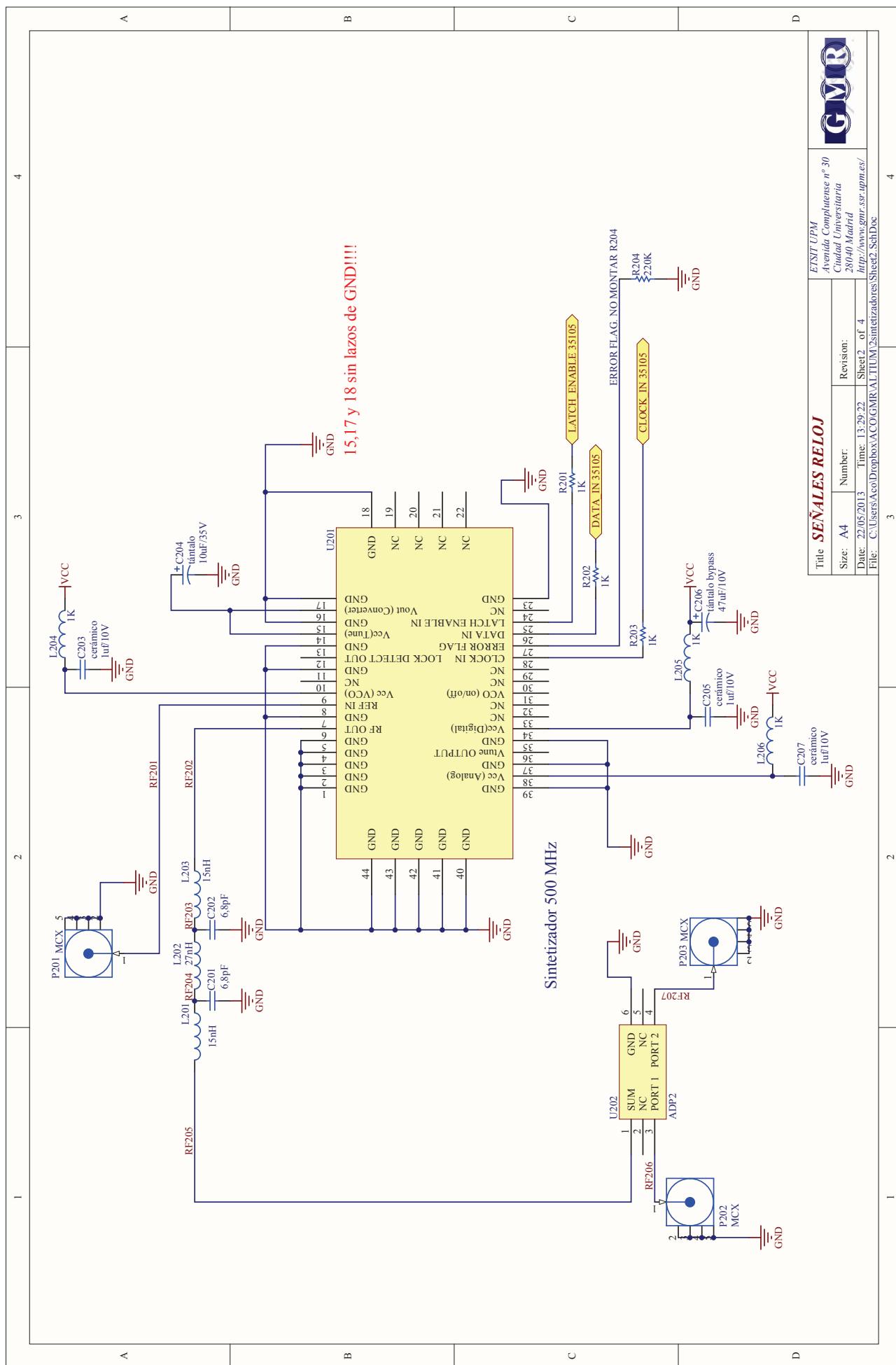
Una posible solución es hacer que el PIC sea capaz de discernir entre una pulsación retenida o larga y una corta del pulsador S1 integrado en el circuito impreso que alberga el PIC. La pulsación corta seguirá teniendo la misma funcionalidad inicial de reset del equipo. Una pulsación retenida durante más de un segundo provocará que el PIC inicie el proceso por el cual determine cuál de las cadenas almacenadas en la memoria del PIC se ha de transmitir al sintetizador en los sucesivos reset del PIC.

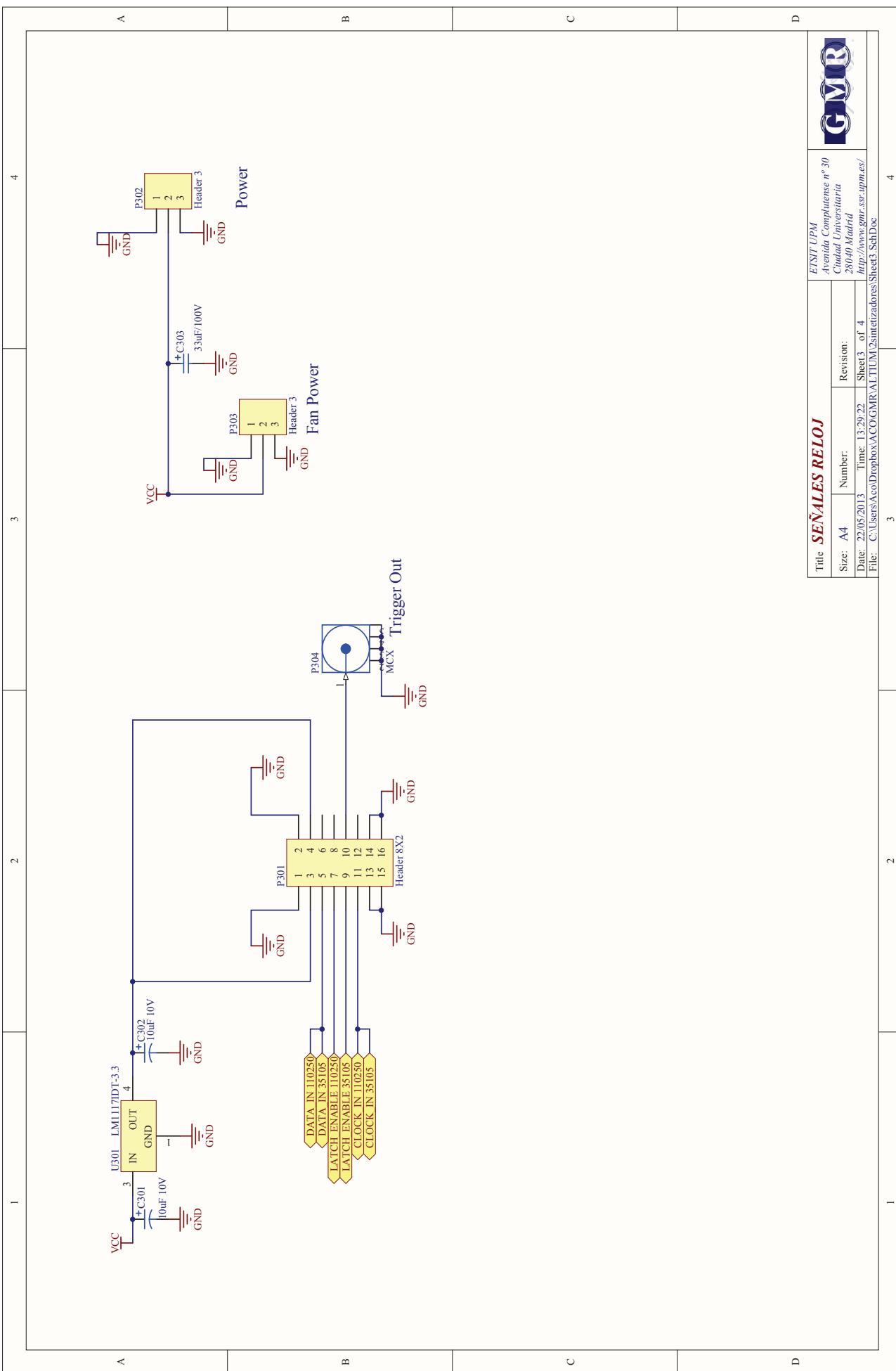
Tras la pulsación larga, la selección de dicha cadena se realizará mediante pulsaciones cortas; así, por ejemplo, para seleccionar la séptima cadena del array de strings definido en el código del programa se deberá pulsar repetidamente siete veces el microinterruptor. Por último, para salir del modo de configuración, guardar los datos en EEPROM y volver al modo de funcionamiento normal, se deberá generar nuevamente una pulsación de más de un segundo de duración.

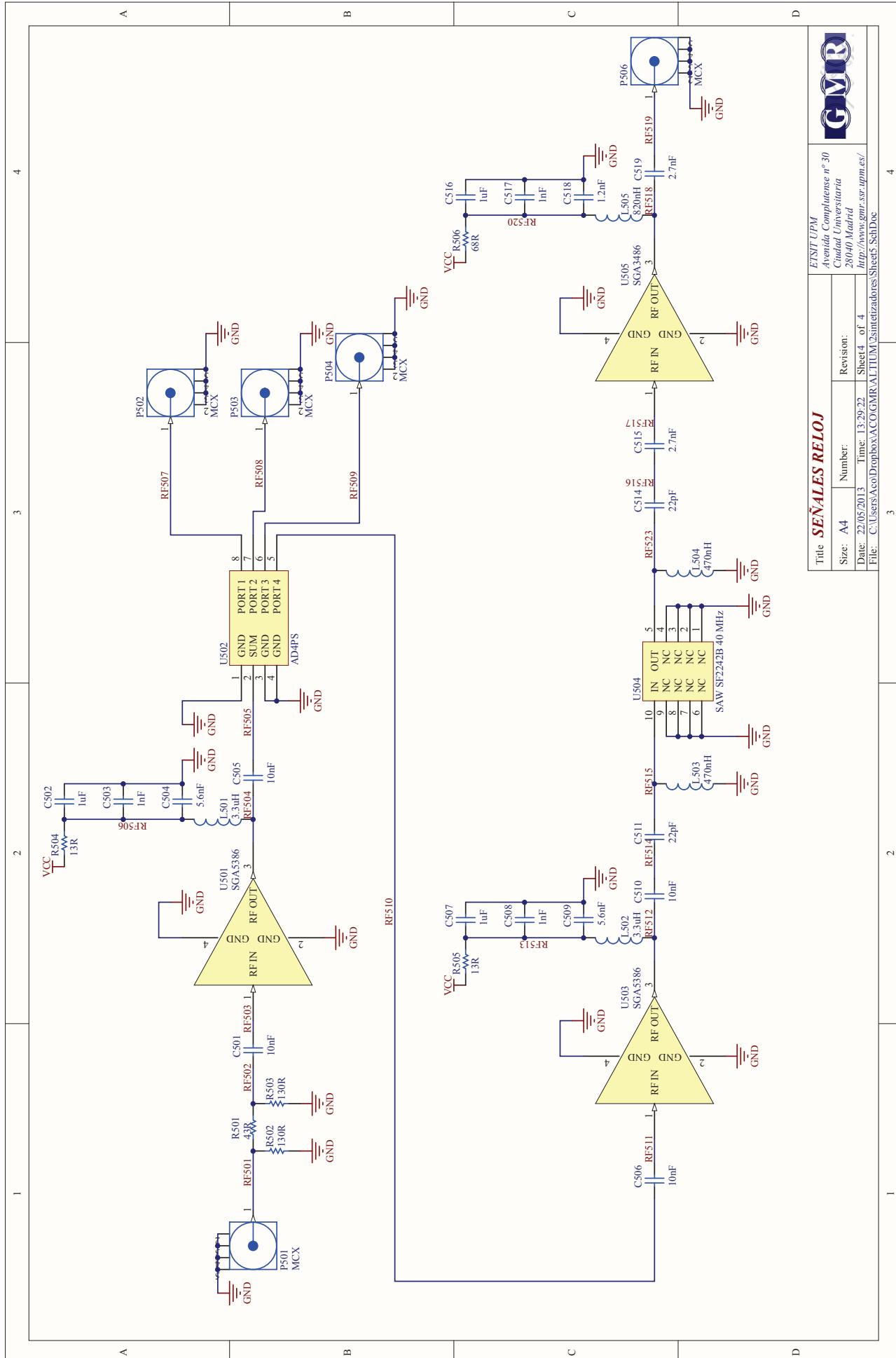
Actualmente se utiliza una única cadena de configuración almacenada en la RAM del PIC. Puesto que la RAM del PIC es limitada y no es capaz de albergar más de una cadena se debería guardar el nuevo array de cadenas en la ROM del PIC así como el índice de la cadena seleccionada en el proceso de configuración en la EEPROM del mismo.

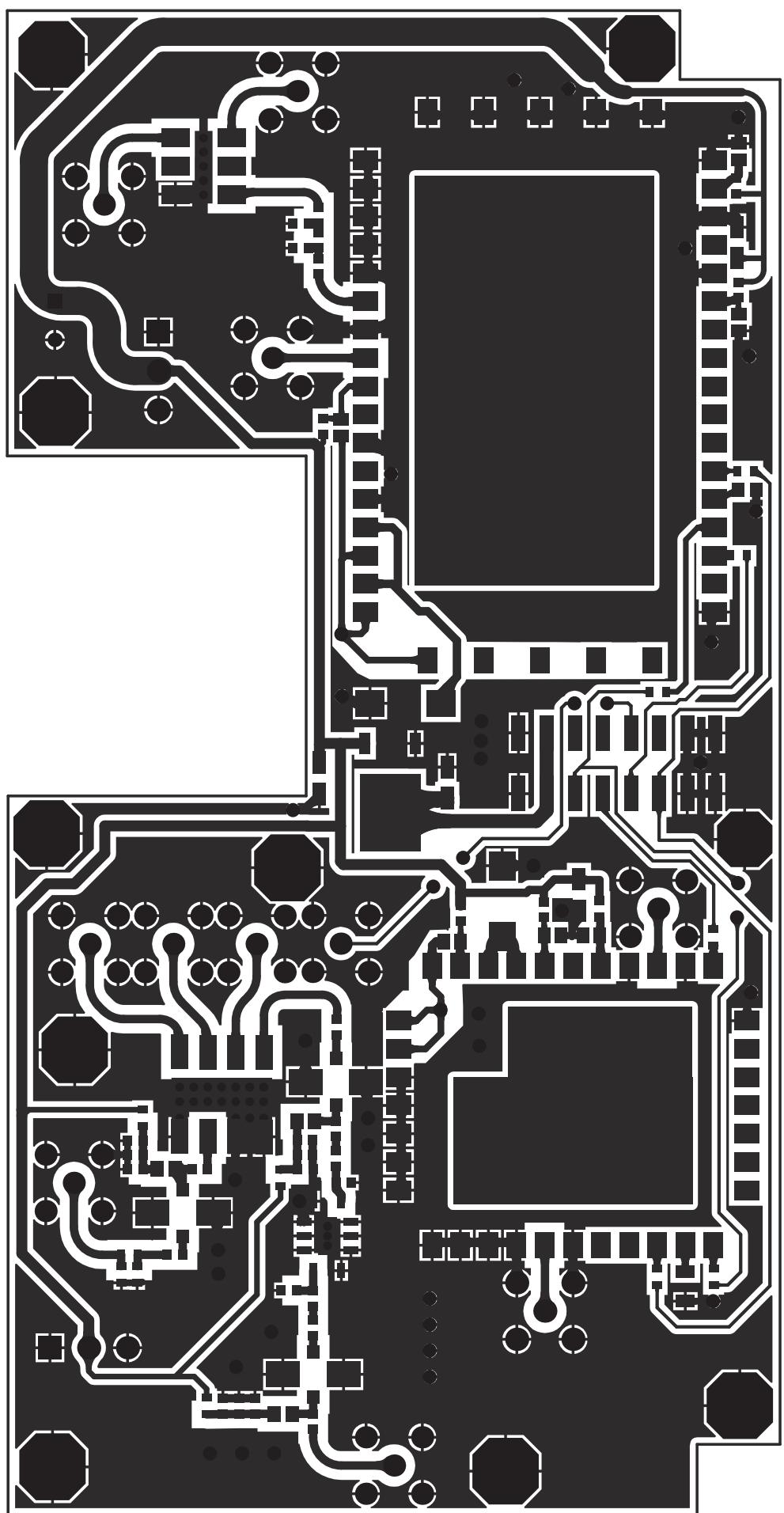
B.3. Esquemas tarjeta de síntesis de señales de reloj.

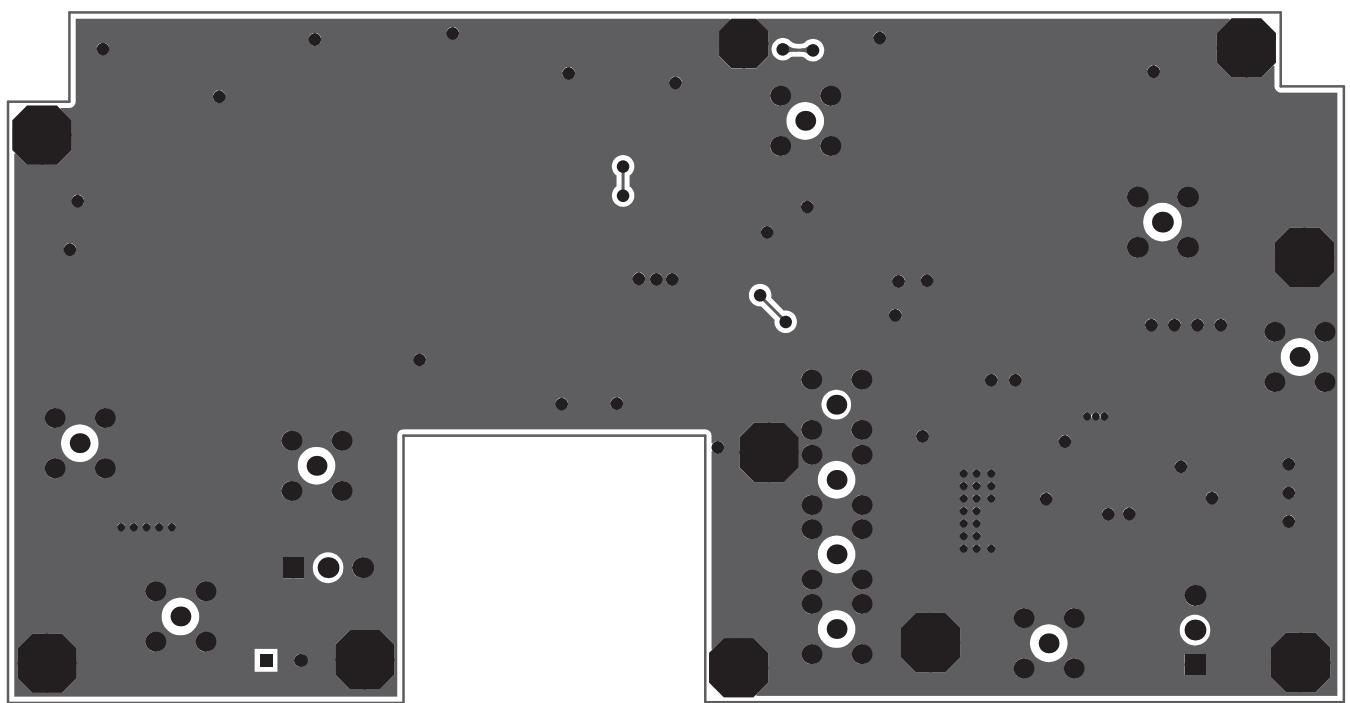


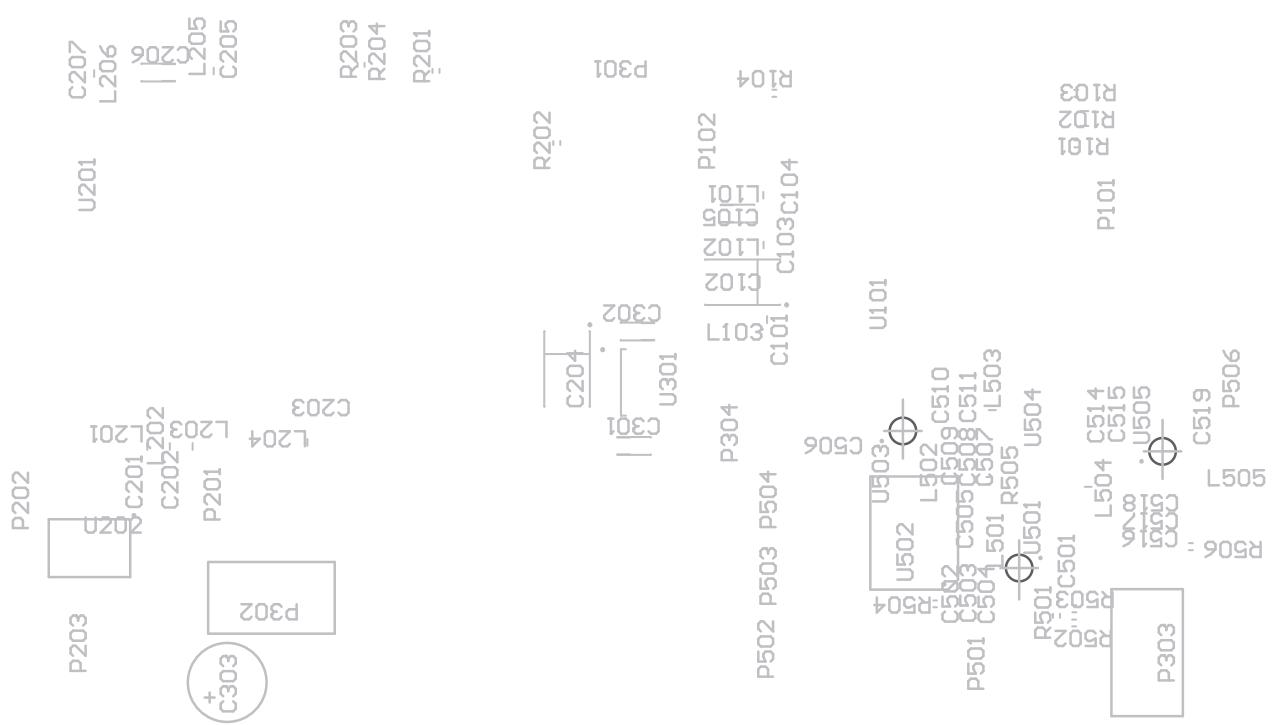


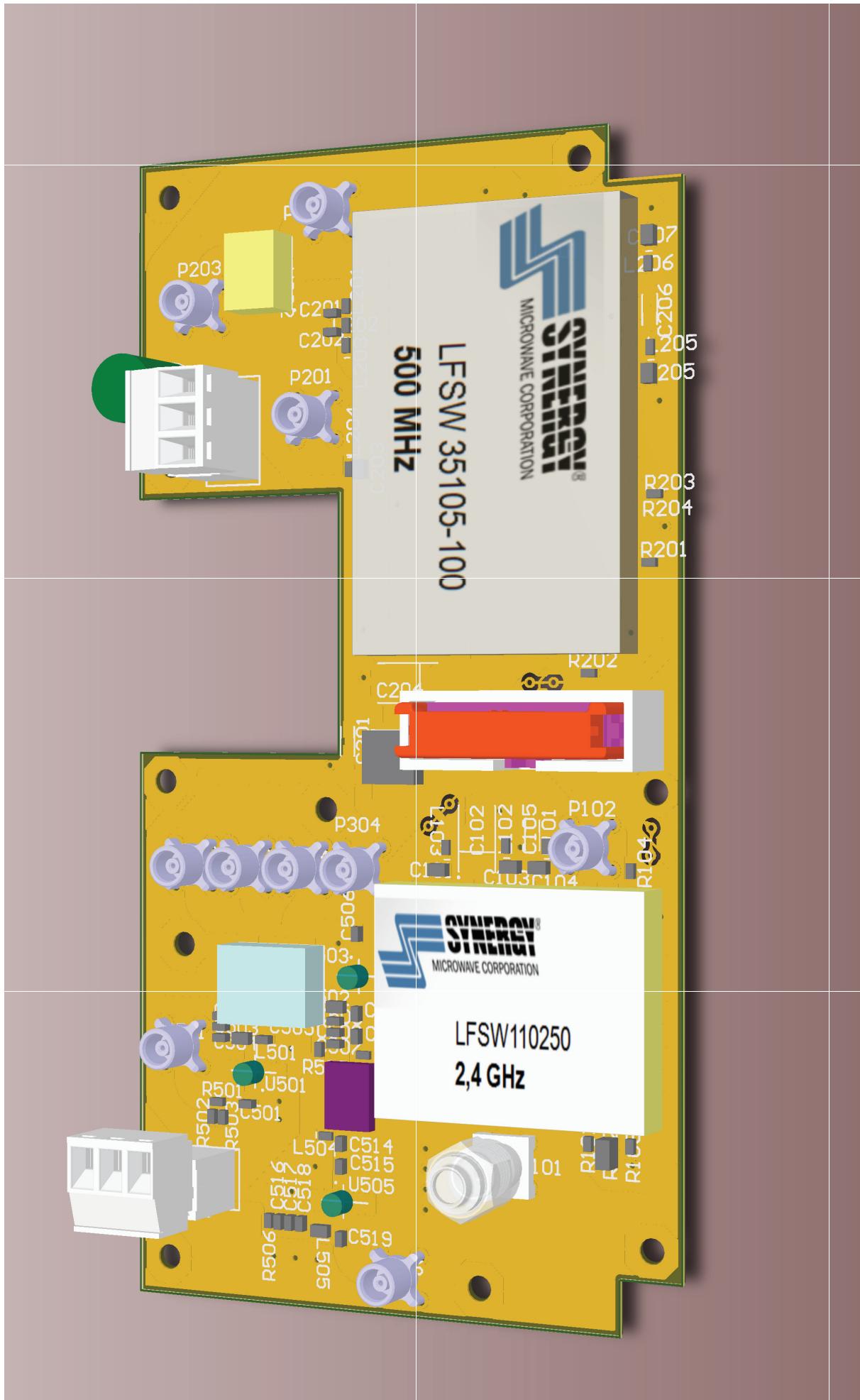




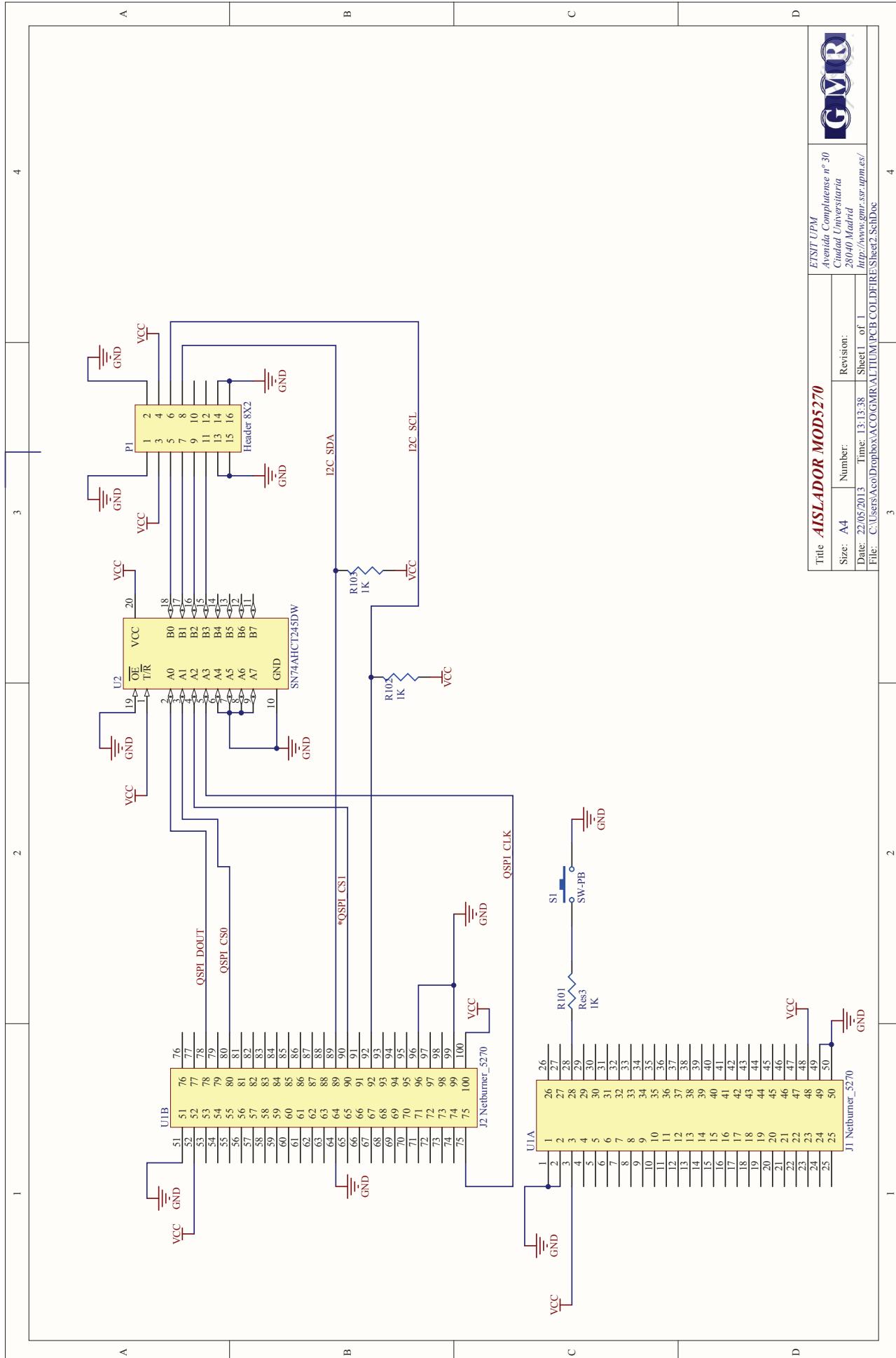




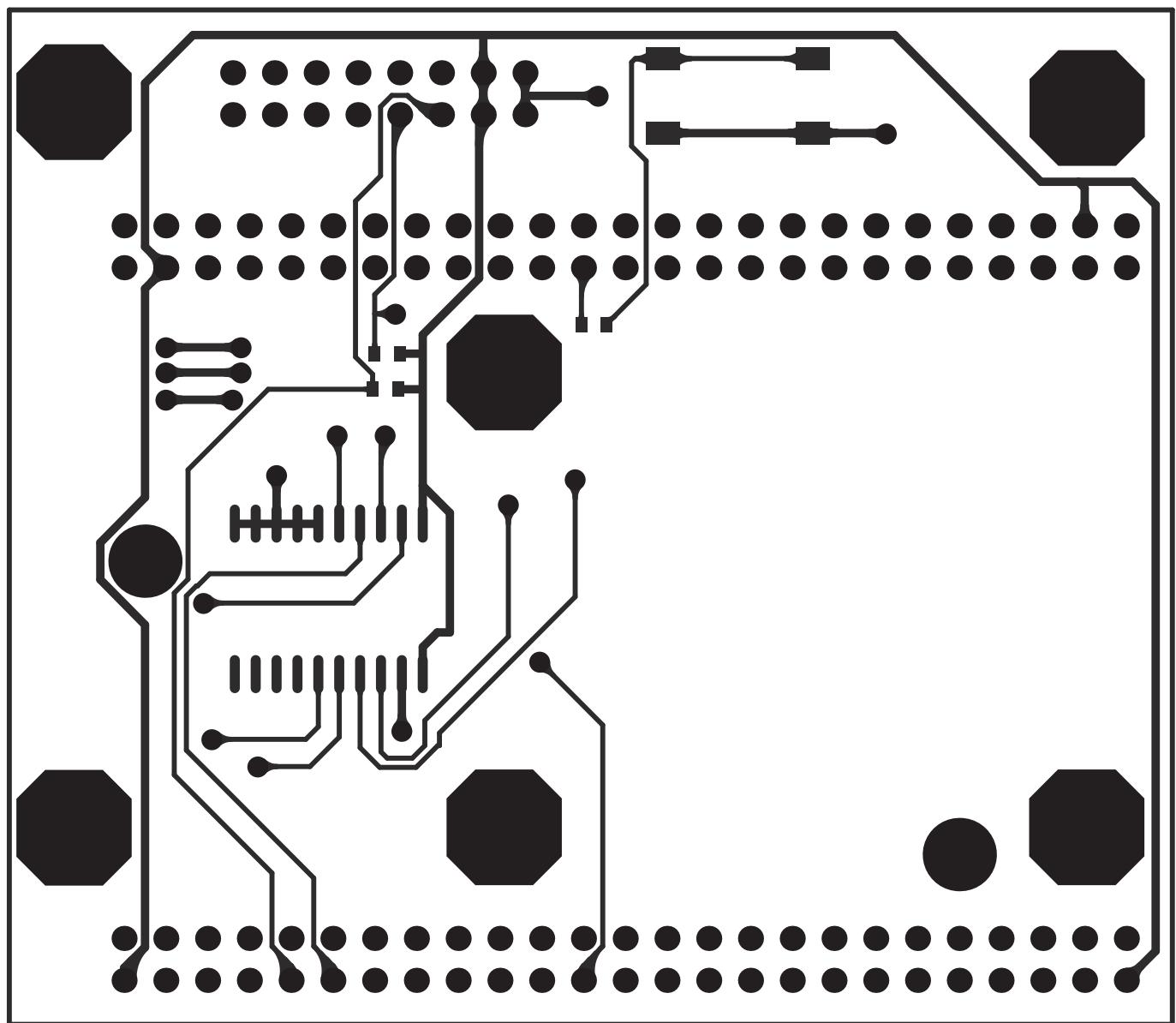


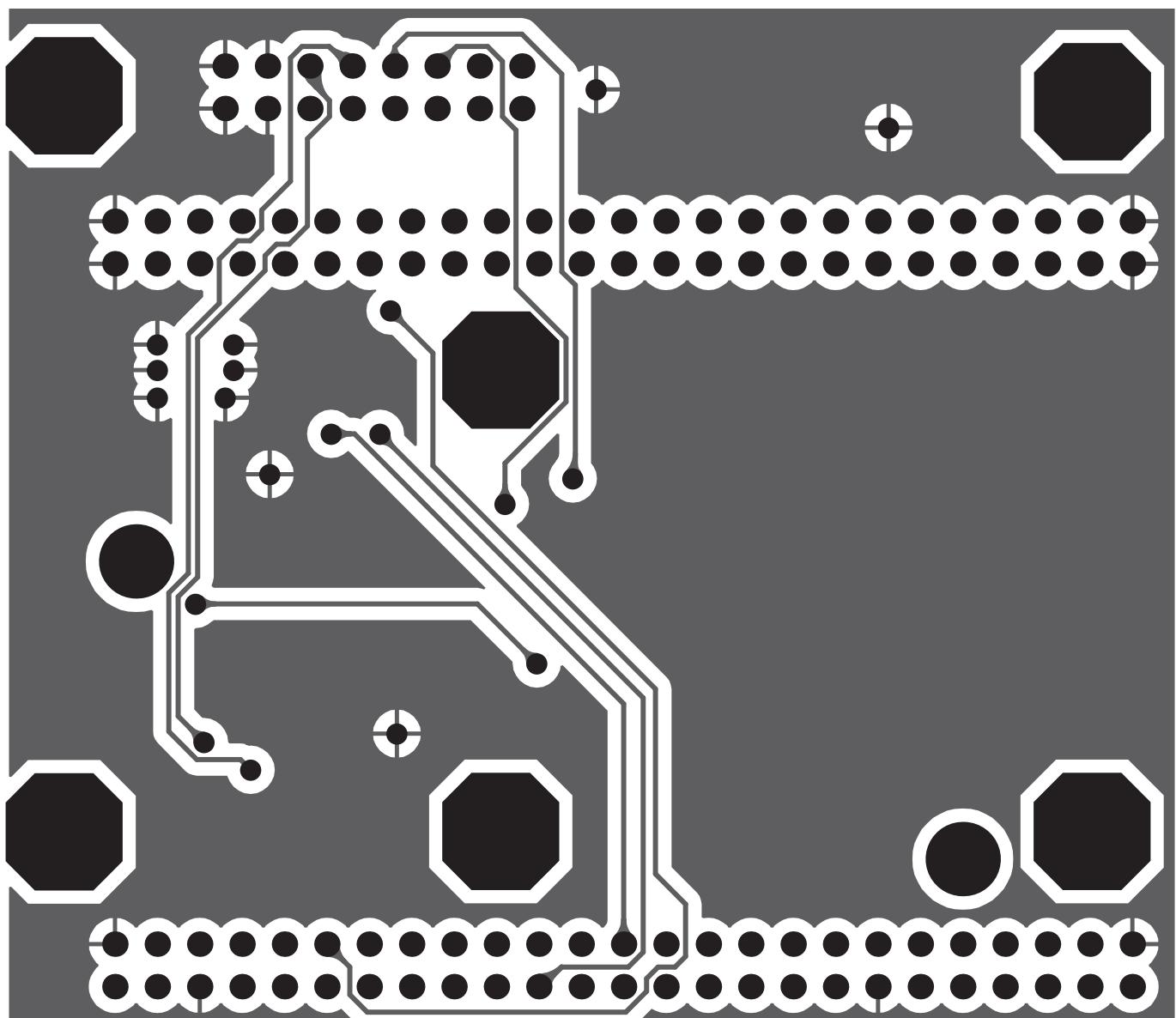


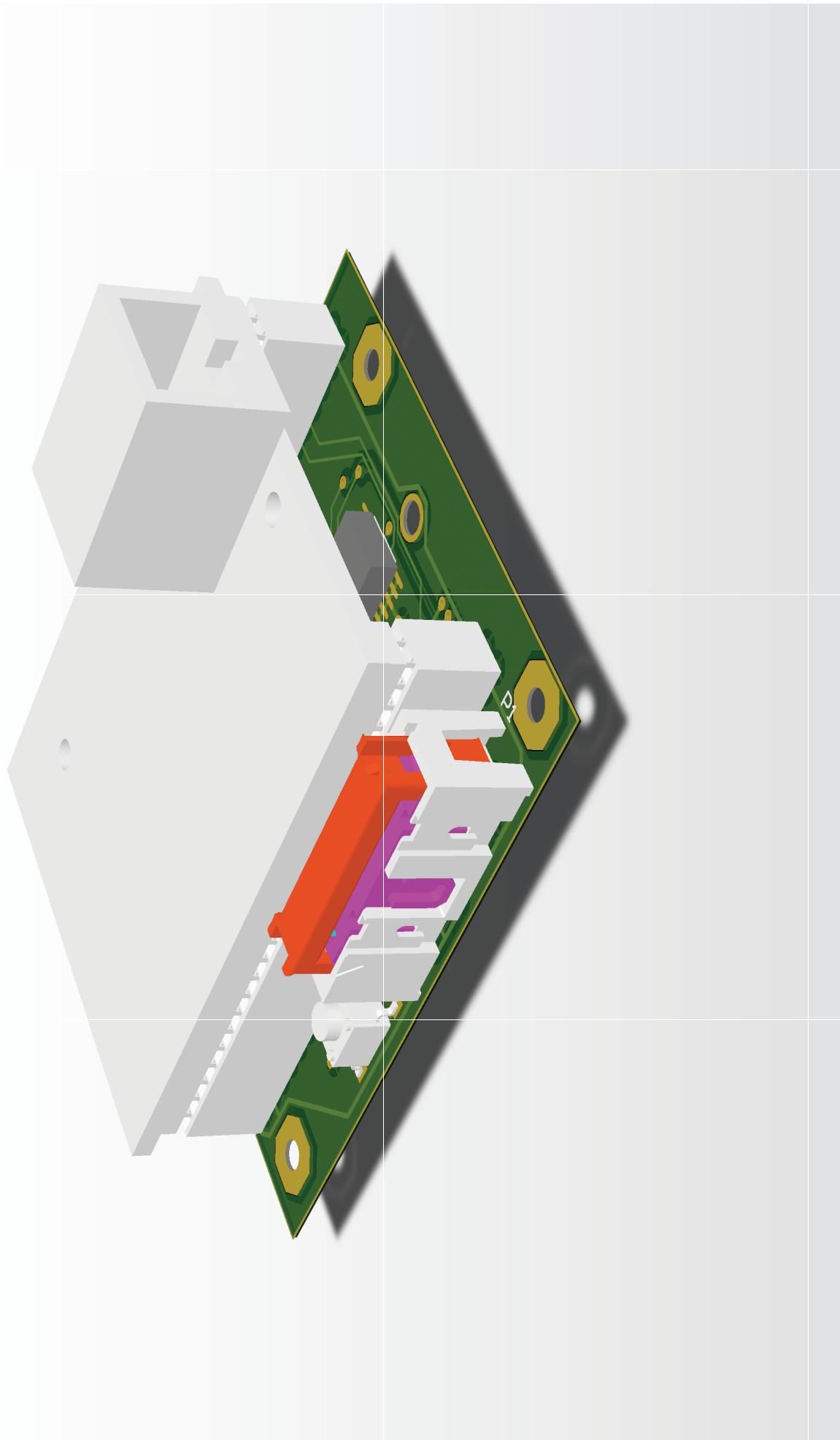
B.4. Esquemas tarjeta acoplamiento MOD5270.



Title AISLADOR MOD5270		EISIT UPM Avenida Complutense nº 30 Ciudad Universitaria 28040 Madrid http://www.grc.ssr.upm.es/	
Size: A4	Number:	Revision:	
Date: 22/05/2013	Time: 13:13:38	Sheet 1 of 1	
File: C:\Users\Ac0\Dropbox\ACO\GMRALIUM\PCBCOLD\FIRESheet2.SchDoc			







B.5. Código tarjeta de síntesis de señales de reloj.

main.cpp

```
#include <NetworkDebug.h>
#include "predef.h"
#include <stdio.h>
#include <ctype.h>
#include <startnet.h>
#include <autoupdate.h>
#include <dhcpcclient.h>
#include <smartrap.h>
#include <taskmon.h>
#include <..\MOD5270\system\sim5270.h>
#include <..\MOD5270\system\gpio5270.h>
#include <cfinter.h>
#include <utils.h>
#include <pins.h>
#include "tipos.h"
#include <string.h>
#include <udp.h>
#include <stdlib.h>
#include <qspi.h> // para poder usar el bus Queue SPI

#define fosc 147456000 // Hz
#define tosc 678168E-8 //us
#define pinSalida GPIO_PIN_DATAL1 // pin 16 J2
#define LATCH_ENABLE GPIO_PIN_DATAL4 // pin 17 J2
#define LATCH_ENABLE2 GPIO_PIN_DATAL5 // pin 19 J2
#define DATA_CLOCK GPIO_PIN_DATAL3 // pin 23 J2
#define DATA GPIO_PIN_DATAL2 // pin 18 J2
//#define ipDestino "192.168.2.2"
#define ipDestino "138.4.36.21"
#define puertoDestino 30001

extern "C"
{
void UserMain( void *pd );
void SetIntc( long func, int vector, int level, int prio );
void ModificaPITR( int pitr_ch );
void generaMensaje(void);
void UdpReaderMain(void * pd);
int calculaPrescaller(void);
void SetUpSintetizador(int sintetizador);
void enviaDatos(void);
}
const char *AppName = "GeneradorTrigger"; // App name for IPSsetup
PMR pmr; // var global
int repeticiones=0; // número de repeticiones
char buffer[80]="holo";
//BYTE mensajeSpi[80]={"4D3030303030393630"}; // cadena configuración LFSW110250 a 2.4 GHz
char sintetizador2_4[40]={"4D30303030393630"};
char sintetizador500[40]={"4D30303030314634"};
//char mensajeSpi500[40]={0x4D,0x30,0x30,0x30,0x30,0x30,0x33,0x45,0x38}; //640MHz
char bytes[19] = "4D30303030393630";
//static BYTE RXBuffer[10000]; //, TXBuffer[10000];
bool clockAlto=true;
// SPI manual
int medioClock=0;
int esperarCiclos=10;
int enviarDatos=0;
int byteEnCurso=0;
int bitDelByte=0;
int nbytes = 18; // numero de bytes a tx.
int dato; // variable local de enviadato. Se hace global para controlar en debug.
BYTE clock=0; // reloj de datos SPI
```

main.cpp

```
int sintetizador2=0;
// Allocate task stack for UDP listen task
DWORD UdpTestStk[USER_TASK_STK_SIZE];

///////////////////////////////
// INTERRUPT - PIT interrupt service routine
//
INTERRUPT( trigger_pitr_func, 0x2600 ){
    bool activa=false;
    WORD tmp = sim.pit[1].pcsr; // Get PIT1 Control & Status Register data
    // Clear PIT1 - Refer to table 21-3 for more information on what
    if (pmr.repeticiones!=0){
        repeticiones++;
        if (repeticiones<=pmr.repeticiones) activa=true;
    }else{
        activa=true;
    }
    if (activa){
        // bits are being cleared and set
        tmp &= 0xFF0F; // Bits 4-7 cleared
        tmp |= 0x0F; // Bits 0-3 set
        sim.pit[1].pcsr = tmp;
        if ( clockAlto) {
            // poner a nivel bajo y configurar timer con toff
            sim.pit[1].pmr=pmr.On;
            sim.gpio.podr_datal &= ~pinSalida;
            clockAlto=false;
        } else {
            sim.pit[1].pmr=pmr.Off;
            sim.gpio.podr_datal |= pinSalida;
            clockAlto=true;
        }
    } // activa
} // end interrupt

/////////////////////////////
// INTERRUPT - PIT interrupt service routine timer 2 para programación
// Sintetizadores (interrumpe cada 1,5 ms)
INTERRUPT( Sintetizadores_pitr_func, 0x2600 ){
    WORD tmp = sim.pit[2].pcsr; // Get PIT1 Control & Status Register data
    // Clear PIT1 - Refer to table 21-3 for more information on what
    // bits are being cleared and set
    tmp &= 0xFF0F; // Bits 4-7 cleared
    tmp |= 0x0F; // Bits 0-3 set
    sim.pit[2].pcsr = tmp;
    sim.pit[2].pmr=55000; // BORRAR LA INTERRUPCIÓN

    medioClock=!medioClock; // cambiamos cada interrupción
    if (!medioClock) {
        // fin de bit. Cambiar reloj
        clock = sim.gpio.ppdtsr_datal & DATA_CLOCK; // leer para cambiar
        if (clock){
            sim.gpio.podr_datal &= ~DATA_CLOCK;
            clock=0;
        }else {
            sim.gpio.podr_datal |= DATA_CLOCK;
            clock=1;
        }
        // decrementar espera
        if (esperarCiclos>0) {
```

```

main.cpp

        esperarCiclos--;
    }
} else {
    // mitad de ciclo alto o bajo
    if (esperarCiclos==0 && clock==0 && !enviarDatos){
        // enviamos datos
        enviarDatos=1;
        if (sintetizador2){
            sim.gpio.podr_datal &= ~LATCH_ENABLE2;
        }else{
            sim.gpio.podr_datal &= ~LATCH_ENABLE;
        }

        //OSTimeDly(40); //delay 100us
        // si estamos en ciclo bajo en mitad bit y con LE enviamos dato
    }
    if (enviarDatos && clock==0) enviaDato();
} // !medioclock
} // end interrupt

///////////////////////////////
// SetUpPITR PIT - setup function. See chapter 21 of the 5271 reference
// manual for details
//
/* table 21-3 in the reference manual for bits 8-11 */
int SetUpPITR( int pitr_ch, WORD clock_interval, BYTE pcsrc_pre){
    WORD tmp;
    if ( ( pitr_ch < 1 ) || ( pitr_ch > 3 ) ){
        iprintf( "*** ERROR - PIT channel out of range ***\r\n" );
        return -1;
    }
    // Populate the interrupt vector in the interrupt controller. The
    // SetIntc() function is supplied by the NetBurner API to make the
    // interrupt control register configuration easier
    //J2[48] = 0;
    SetIntc( ( long ) &trigger_pitr_func, 36 + pitr_ch, 2 /* IRQ2 */, 3 );
    // iniciamos timer con tOn
    // Configure the PIT for the specified time values
    sim.pit[pitr_ch].pmr = pmr.On; // Set PIT modulus value
    tmp = pmr.prescaler;
    tmp = ( tmp << 8 ) | 0x0F;
    sim.pit[pitr_ch].pcsr = tmp; // Set system clock divisor
    // set bits [3:0] in PCSR
    return 0;
}
// genera el mensaje ack con los datos calculados
void generaMensaje(void){
    char buf[80]="Parametros del timer: PMRon=";
    char buf2[10];
    char buf3[10]=" PMRoff=";
    char buf4[14]=" Prescaller=";
    char *f=buf2;
    UDPPacket pkt;
    IPADDR ipaddr;
    // añadimos pmrOn
    itoa(pmr.On,f ,10);
    strcat(buf,f);
    // texto
    f=buf3;
    strcat(buf,f);
    // pmroff
}

```

```

main.cpp

itoa(pmr.Off,f ,10);
strcat(buf,f);
// texto
f=buf4;
strcat(buf,f);
// añadimos prescaller
itoa(pmr.prescaller,f ,10);
strcat(buf,f);
f=buf;
//enviar ack por udp
pkt.SetSourcePort(puertoDestino); //usamos el mismo para rx y tx
pkt.SetDestinationPort(puertoDestino);
pkt.AddData(f);
pkt.AddDataByte(0);
ipaddr = AsciiToIp(ipDestino);
pkt.Send(ipaddr);
iprintf( "Enviado: " );
iprintf( (char*)f );
iprintf( "\r\n");
}

///////////////
// UserMain
//
void UserMain( void *pd ){
    int portnum=30001;
    InitializeStack();
    // inicializar debub por red cuando sea necesario
    #ifdef _DEBUG
        InitializeNetworkGDB_and_Wait();
    #endif // _DEBUG
    OSChangePrio( MAIN_PRIO );
    EnableAutoUpdate();
    OSTimeDly(10);
    iprintf( "Aplicación iniciada\r\n" );
    sim.gpio.pddr_datal |= pinSalida;
    sim.gpio.pddr_datal |= LATCH_ENABLE;
    sim.gpio.pddr_datal |= LATCH_ENABLE2;
    sim.gpio.pddr_datal |= DATA;
    sim.gpio.pddr_datal |= DATA_CLOCK;
    SetUpSintetizador(1); // aprox 1,5 ms
    OSTimeDly(40);
    SetUpSintetizador(2); // aprox 1,5 ms
    OSTimeDly(40);
    // Create UDP listen task
    // higher priority than send task
    OSTaskCreate(UdpReaderMain,(void *)portnum,&UdpTestStk[USER_TASK_STK_SIZE]
,UdpTestStk,MAIN_PRIO - 1);
    while ( 1 ){
        //iprintf( "Vivo\r\n" );
    }
} // end usermain

void UdpReaderMain(void * pd){
    int port = (int)pd; // cast void * param as int port number
    char buffer2[8];
    char buffer3[8];
    char buffer4[16];
    char c=':';
    char *buf;
    char *c2,*c3,*c4,*c5;
    int len;
}

```

```

main.cpp

iprintf("Listening on port: %d\r\n", port);
// Create a FIFO for the UDP packet and initialize it
OS_FIFO fifo;
OSFifoInit(&fifo);
// Register to listen for UDP packets on port number 'port'
RegisterUDPFifo(port, &fifo);
while (1){
    UDPPacket upkt(&fifo, 3 * TICKS_PER_SECOND);
    // Did we get a valid packet, or just time out?
    if (upkt.Validate()){
        // obtener pmr.on
        buf=(char*)upkt.GetDataBuffer();
        len=upkt.GetDataSize();
        buf[len] = '\0';
        iprintf("Recibido: ");
        iprintf((char*) buf);
        iprintf("\n\r");
        c2=strchr(buf,c);
        len=c2-buf;
        buffer2[0] = '\0';
        c3=buffer2;
        strncpy(c3,buf,len);
        c3[len]='\0';
        pmr.On=0;
        pmr.On=atoi(c3);
        // obtener pmr.off
        c3=strchr(c2+1,c);
        len=c3-c2;
        buffer3[0] = '\0';
        c4=buffer3;
        strncpy(c4,c2+1,len-1);
        c4[len-1]='\0';
        pmr.Off=atoi(c4);
        // obtener repeticiones
        len=strlen(buf);
        len=buf+len-c3;
        buffer4[0] = '\0';
        c5=buffer4;
        strncpy(c5,c3+1,len-1);
        c5[len-1]='\0';
        pmr.repeticiones=atoi(c5);
        calculaPrescaller();
        generaMensaje(); // envia por udp configuracion timer
        SetUpPITR(1 , pmr.On, pmr.prescaller);
        iprintf("Iniciado timer\r\n");
    } //end if
} // end while
} // end udpreadermain

int calculaPrescaller(void){
    int tOn,tOff,tmin,tmax;
    // valores por defecto
    tOn=(int)pmr.On;
    tOff=(int)pmr.Off;
    (tOn>tOff)? tmin=tOff :tmin=tOn; // establecer el tiempo mínimo
    (tOn>tOff)? tmax=tOn :tmax=tOff; // establecer el tiempo máximo
    // calcular el prescaller necesario
    if (tmin<15){
        //enviaUdp("Error. Valor mínimo > 15 us");
        return -1;
    }
}

```

main.cpp

```
int prescaller=16;
double tCuenta;
// obtener el menor prescaller que permita tmax
if (tmax<=29126667) prescaller=15;
if (tmax<=14563333) prescaller=14;
if (tmax<=7281667) prescaller=13;
if (tmax<=3640833) prescaller=12;
if (tmax<=1820417) prescaller=11;
if (tmax<=910208) prescaller=10;
if (tmax<=455104) prescaller=9;
if (tmax<=227552) prescaller=8;
if (tmax<=113776) prescaller=7;
if (tmax<=56888) prescaller=6;
if (tmax<=28444) prescaller=5;
if (tmax<=14222) prescaller=4;
if (tmax<=7111) prescaller=3;
if (tmax<=3556) prescaller=2;
if (tmax<=1778) prescaller=1;
if (tmax<=889) prescaller=0;
if (prescaller==16){
    iprintf( "*** Error Máximo tOn=29,12 segundos ***\r\n" );
    return -1;
}
tCuenta=2*tosc; //us
for (int n=0;n<prescaller;n++){
    tCuenta=tCuenta*2; // bucle equivale a tcuenta=tCuenta*2^prescaller;
}
pmr.On=tOn/tCuenta;
pmr.Off=tOff/tCuenta;
pmr.prescaller=prescaller;
pmr.On-=1;
pmr.Off+=5;
#ifndef _DEBUG // si programamos timer corto no podemos hacer debug por ethernet
    if ((pmr.On<15000||pmr.Off<15000) && prescaller<2){
        pmr.On=30000;
        pmr.Off=60000;
        pmr.prescaller=1;
        iprintf( "Para compatibilidad con debug via ethernet tOn=32256 y tOff=2*tOn\r\n" );
    }
#endif // _DEBUG
return 0; // salida sin error
}

void SetUpSintetizador(int sintetizador){
    WORD tmp;
    bytes[0]='\0';
    if (sintetizador==1){
        strcat(bytes,sintetizador2_4);
    }else{
        strcat(bytes,sintetizador500);
        sintetizador2=1; // para activar latch_enable2
    }
    esperarCiclos=10;
    byteEnCurso=0;
    bitDelByte=0;
    enviarDatos=0;

    // Programar timer reloj SPI
    SetIntc( ( long ) &Sintetizadores_pitr_func, 36 + 2, 2 /* IRQ2 */, 3 );
    sim.pit[2].pmr = 55000; // Set PIT modulus value
    tmp = 1;
```

```

main.cpp

tmp = ( tmp << 8 ) | 0x0F;
sim.pit[2].pcsr = tmp; // Set system clock divisor
}

// envia un bit de datos. Si termina secuencia modifica esperarCiclos
// para que la IRQ desactive LE
void enviaDato(void) {
    if (bitDelByte==4) {
        // cambiar de byte
        byteEnCurso++;
        bitDelByte=0;
    }
    if (byteEnCurso==nbytes){
        // se termino de enviar datos
        if (sintetizador2){
            sim.gpio.podr_datal |= LATCH_ENABLE2;
        }else{
            sim.gpio.podr_datal |= LATCH_ENABLE;
        }

        // sim.pit[2].pmr=5500; // desactivar timer2
    } else{
        // enviar datos
        // en SPI se transmiten los bytes por orden y los bits de mas peso a menos
        dato=bytes[byteEnCurso];
        if (dato>57){
            // es letra A-F A=65-55=10 (VER ASCII)
            dato-=55;
        } else {
            // es numero 0-9 0=48-48=0 (VER ASCII)
            dato-=48;
        }
        if (bitDelByte==0) dato=dato&0x8;
        if (bitDelByte==1) dato=dato&0x4;
        if (bitDelByte==2) dato=dato&0x2;
        if (bitDelByte==3) dato=dato&0x1;
        if (dato>0){
            sim.gpio.podr_datal |= DATA;
        } else {
            sim.gpio.podr_datal &= ~DATA;
        }
        bitDelByte++;
    } // end if
} // enviadato

```


Apéndice C

DDS Analog Devices.

El DDS de la marca Analog Devices es el modelo AD9958. En la figura C.1 se muestra integrado dentro de la placa de evaluación AD9958/PCBZ que pone a disposición el fabricante.

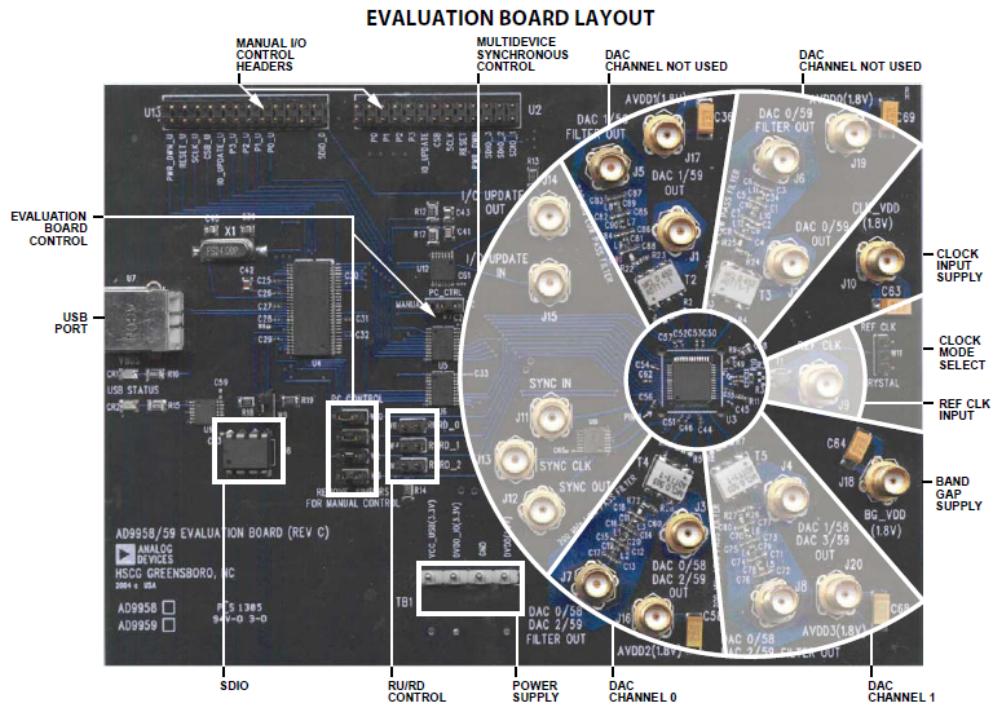


Figura C.1: Placa de evaluación AD9958/PCBZ.

El circuito integrado incorpora dos DDS de 500 MSPS. Cada uno de ellos dispone de un acumulador de fase de 32 bits y un conversor digital analógico de 10 bits.

El DDS es capaz de generar un único tono o realizar un barrido lineal o arbitrario en frecuencia. Para un barrido lineal, como el mostrado en la figura C.2, se deben configurar

los registros de 32 bits indicados en la tabla C.1.

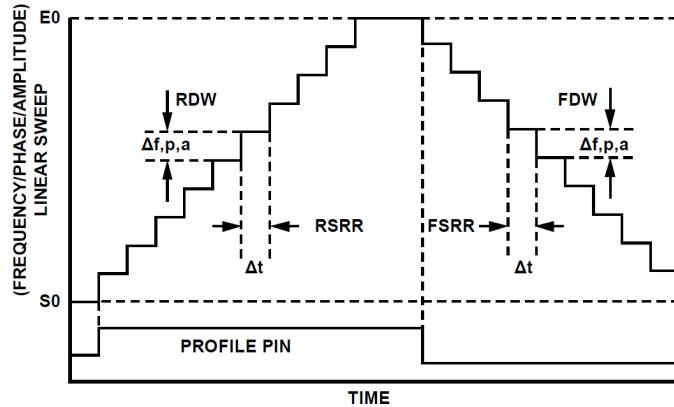


Figura C.2: Forma de onda del barrido lineal en AD9885.

Registro	Descripción
S0 (start point)	Frecuencia inicial del barrido
E0 (end point)	Frecuencia final del barrido
RDW (Rising Delta Word)	Valor de incremento en la rampa de subida
RSRR (Rising Sweep Ramp Rate)	Duración del intervalo en rampa de subida (ciclos de reloj)
FDW (Falling Delta Word)	Valor de decremento en la rampa de bajada
FSRR (Falling Sweep Ramp Rate)	Duración del intervalo en rampa de bajada (ciclos de reloj)

Tabla C.1: Parámetros del barrido lineal en frecuencia.

El incremento de frecuencia Δf viene determinado por la fórmula C.1, donde $SYS-CLOCK$ se expresa en Hz. La duración del escalón de la rampa de subida Δt viene determinada por la fórmula C.2. Las mismas fórmulas son aplicables para la rampa de bajada.

$$\Delta f = \frac{RDW}{2^{32}} \cdot SYS-CLOCK \quad (C.1)$$

$$\Delta t = RSRR \cdot \frac{4}{SYS-CLOCK} \quad (C.2)$$

Cuando se inicia el barrido un temporizador ajustable de 8 bits se carga con el tiempo de subida deseado. Durante ese tiempo la frecuencia aumenta según lo descrito anteriormente. Cuando el contador vale 1 se vuelve a cargar dicho contador con el tiempo deseado de rampa de bajada y se cargan los valores FDW y FSSR. Opcionalmente se puede configurar el DDS en modo NO_DWELL_MODE. En este modo, cuando el contador vale 1, se carga de nuevo el valor S0 con los valores RDW y RSSR. De tal forma, el DDS produce una forma de onda de diente de sierra.

Además, se puede tener un control hardware del inicio y final de rampa a través de la entrada PROFILE_PIN mediante una señal externa LVCMOS (*Low Voltage Complementary Metal Oxide Semiconductor*) como se muestra en la figura C.2. En los flancos de subida de la señal externa, el sistema carga los parámetros de subida de rampa (RDW y RSSR). En los flancos de bajada el DDS carga los parámetros FDW y FSSR.

La placa de evaluación dispone de una *GUI* mostrada en la figura C.3 para la configuración de la misma mediante un puerto USB.

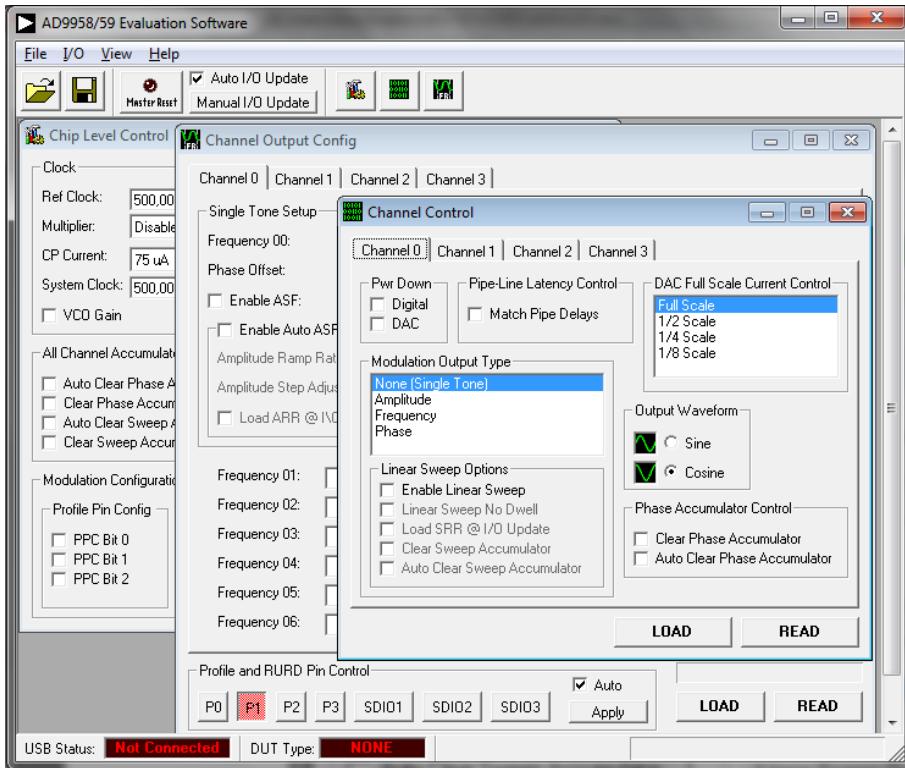


Figura C.3: *GUI* de la placa de evaluación AD9958/PCBZ.

La placa de evaluación del DDS se integra dentro de un diseño realizado en colaboración con... (Pablo Dorta + ref) que se muestra en la figura C.4.

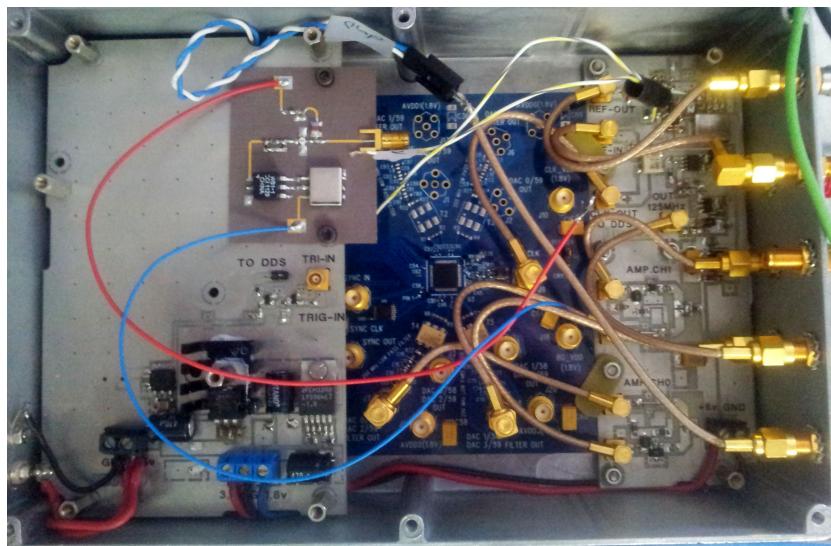


Figura C.4: *DDS AD9958*.

En este diseño, la entrada de reloj del DDS está conectada a una de las réplicas de

la salida del sintetizador de 500 MHz (véase el apartado 5.1.1.5). A la salida del DDS se acopla un PLL que multiplica la frecuencia de la señal generada por el DDS. Para que el PLL no pierda el control del lazo, no deben existir variaciones bruscas de frecuencia; Por ello, cuando el DDS alcanza la mayor frecuencia a generar, éste debe retornar a la frecuencia de inicio de rampa con una pendiente suficiente como para no perder el lazo.

Este diseño necesita de una señal externa de *trigger* para determinar el instante de inicio de subida y el de bajada de cada rampa. Por ello, parte de este proyecto contempla la fabricación de un generador de señal de disparo que se describe en el apartado C.1.

El diseño de Pablo Dorta incluye dos puertas inversoras incorporadas tras el conector MCX que introduce la señal de *trigger* externo. Una posible razón para tal implementación es la de añadir un retardo a la señal de *trigger*. En la aplicación final del radar en la cual se pretende integrar este sistema de generación dicho retardo no tiene ninguna utilidad. Por ello, se ha decidido retirar la doble puerta inversora. Una vez realizado el *bypass* a las puertas inversoras se ha comprobado que la señal de trigger emitida por el generador de *trigger* descrito en C.1 no es capaz de excitar el DDS para que trabaje correctamente.

El DDS, en este sintetizador de señal, ha sido configurado para que utilice como pin de control de subida y bajada de rampa el PROFILE_PIN denominado P1. Tras introducirse la señal externa de trigger en el conector MCX y atravesar la doble puerta inversora (ahora puenteadas), la señal se introduce en la placa de evaluación a través del pin 5 del conector U2 que se visualiza en la parte superior de la figura C.1 y cuyo esquema se muestra en la figura C.5

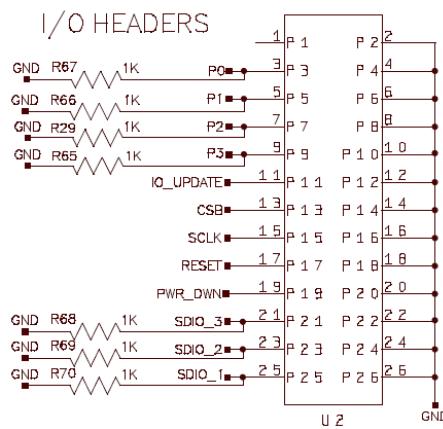


Figura C.5: Esquema del conector U2 de la placa de evaluación.

Según se muestra en el esquema de la figura C.6, la misma pista del pin 5 del conector U2 confluye con el pin de salida Y5 del buffer U5. Dicho buffer sirve para que la aplicación GUI pueda generar un trigger interno que controle el sentido de la rampa mediante la señal P1_U en lugar de una señal de *trigger* externa .

El buffer U5 solo debe activar sus salidas cuando la señal PC_CTRL está a nivel bajo;

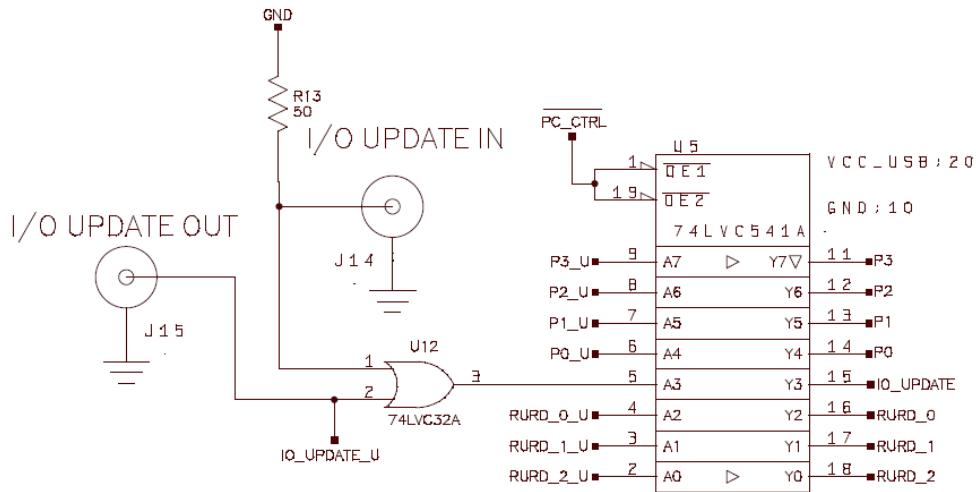


Figura C.6: Buffer de salida de la tarjeta de evaluación.

lo cual es indicativo de que la aplicación GUI tiene control sobre las señal P1 de entrada al DDS. En el caso contrario, debe configurar éstas en estado de alta impedancia.

Se ha comprobado que el buffer no funciona correctamente, ya que, cuando la señal PC_CTRL está a nivel bajo la impedancia del pin Y5 no es alta. A pesar de ello, el buffer funciona correctamente cuando la señal PC_CTRL está a nivel alto. El fallo en el circuito de salida del buffer puede haberse debido a que, en algún momento de uso de la placa donde estaba configurado un funcionamiento mediante trigger interno, haya sido introducida al mismo tiempo una señal de trigger externo.

A pesar de esta avería, cuando la señal trigger se ve reforzada por la acción de la doble puerta inversora, cuya impedancia de salida es menor que la de salida del generador trigger, el DDS detecta los pulsos correctamente.

Como la aplicación final no requiere una señal de *trigger* interno, la solución adoptada ha sido la de levantar la pata de salida Y5 de su pista. De esta manera la señal externa por sí sola es capaz de excitar el DDS.

La decisión de anular la doble puerta inversora, los trabajos para llevarlo a cabo y la detección del malfuncionamiento en estas condiciones no forman parte de las tareas realizadas en este proyecto. Sí forma parte del proyecto las tareas realizadas para detectar la razón del malfuncionamiento y tratar de buscar una solución al problema.

C.1. Generador de señal de disparo.

Según se ha descrito en el apartado anterior, la generación de la señal chirp mediante el DDS de Analog Devices requiere de un *trigger* periódico y configurable. Dicho *trigger*

es generado por el microcontrolador incorporado en el módulo MOD5270 (véase B.1) el cual se ubica en la caja de síntesis (véase 5.1.1.6).

La señal de *trigger* se genera programando uno de los 4 temporizadores programables que dispone el microcontrolador cuya arquitectura se muestra en la figura C.7. Inicialmente se activa la señal de *trigger* y se programa el timer. Una vez alcanzado el valor de la cuenta se genera una interrupción de máximo nivel. En la rutina de atención de la interrupción se reprograma el timer para configurar el tiempo en el que la señal ha de estar inactiva.

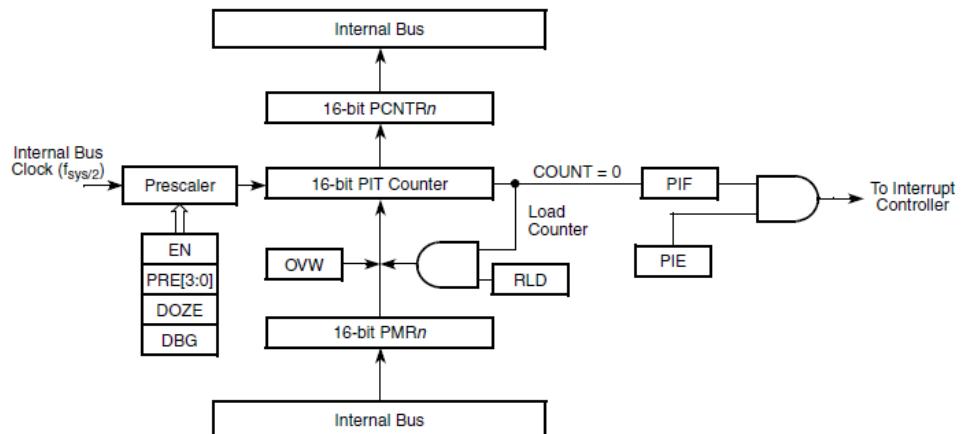


Figura C.7: Diagrama de bloques de los temporizadores programables.

La señal de *trigger* se transmite por el pin *PDATAH0* del bus de datos *GPIO* (*General Purpose Input/Output*). La PCB en la que se acopla el microcontrolador conduce la señal hasta su conector IDC de 16 pines. La señal llega a la tarjeta de síntesis por la cinta plana que une el microcontrolador y dicha tarjeta. La señal termina finalmente en un conector MCX identificado en la tarjeta como P304 según muestra la figura C.8.

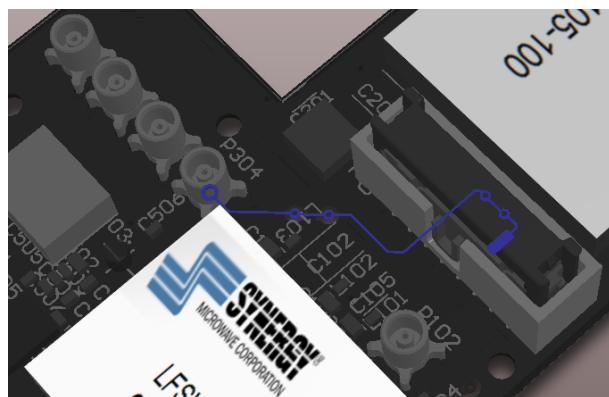


Figura C.8: Conector de salida de la señal de *trigger*.

La configuración de los parámetros del *timer* se ajustan mediante la recepción de un mensaje UDP enviado a la dirección IP *138.4.36.23*. El formato del mensaje ha de ser *1:AAAA:BBBB:CCCC*, donde 1 identifica que el mensaje contiene información para la configuración del timer, el campo *AAAA* indica el tiempo que el *trigger* ha de estar activo, *BBBB* idem de *AAAA* pero para el estado inactivo y *CCCC* indica el número de repeticiones. Todos los números se han de escribir en formato hexadecimal y la base de tiempos es el μs . Un número de repeticiones nulo implica una señal periódica.

Una vez recibido el mensaje el programa analiza cuál es el menor *prescaler* que se puede configurar para permitir que el periodo más largo se pueda codificar en el registro de 16 bits *PCNTR*. Partiendo de este dato el programa calcula el valor que se va a cargar en *PMR* para cada uno de los estados del *trigger* y retorna un mensaje UDP al remitente del mensaje de configuración con el formato *1:AAAA:BBBB:CCCC* donde *AAAA* indica el *prescaler* configurado, *BBBB* el valor de *PMR* para el tiempo activo de la señal *trigger* y *CCCC* el valor de *PCNTR* para el tiempo inactivo.

Para la generación de una señal periódica cuya duración del pulso a nivel alto es de $1470 \mu\text{s}$ y de $30 \mu\text{s}$ a nivel bajo, el microcontrolador ha de recibir el mensaje "1:5BE:1E:000". El periodo del reloj de bus es $\frac{2}{f_{clock}} = 13.6\text{ns}$. El microcontrolador calculará la mínima base de tiempos que garantice una cuenta sin desbordamiento para el intervalo de tiempo mayor. En el caso de configurar el valor del *prescaler* a 1, el máximo valor de la cuenta sin desbordamiento podrá ser $2^{16} \cdot 13.6 = 892\mu\text{s}$ y, por lo tanto, insuficiente. Un *prescaler* con valor 2 permitirá una temporización máxima de $892 \cdot 2 = 1780\mu\text{s}$. Con este valor de *prescaler*, se calcula la duración de los intervalos según las ecuaciones C.3 y C.4.

$$PCNTR_{1470\mu\text{s}} = \frac{1,470,000}{13.6 \cdot 2} = 54023 = D307 \quad (\text{C.3})$$

$$PCNTR_{30\mu\text{s}} = \frac{30,000}{13.6 \cdot 2} = 1102 = 044E \quad (\text{C.4})$$

Finalmente el microcontrolador envía un mensaje de respuesta con la siguiente cadena "1:D307:044E:000". La duración del pulso del *trigger* y el periodo de la señal determinan el tiempo de retrazado y la PRF de la señal generada. Dichos parámetros se han de variar con bastante frecuencia dependiendo del tipo de captura que se quiera realizar. Por esta razón se ha desarrollado una aplicación en Visual C++ que permite configurar de forma sencilla la señal de *trigger* según se muestra en la figura C.9.

La aplicación dispone de 3 pestañas. En la primera pestaña se puede configurar el *trigger* indicando a la aplicación la cantidad de μs que el pulso ha de permanecer a nivel alto (Tiempo ON) y el tiempo que ha de permanecer a nivel bajo (Tiempo OFF). Además se puede indicar el número de repeticiones que se desea que se repita la señal.

La segunda pestaña permite definir también el número de repeticiones. La señal se caracteriza en esta pestaña definiendo el periodo de la misma y su *duty cycle*. La tercera

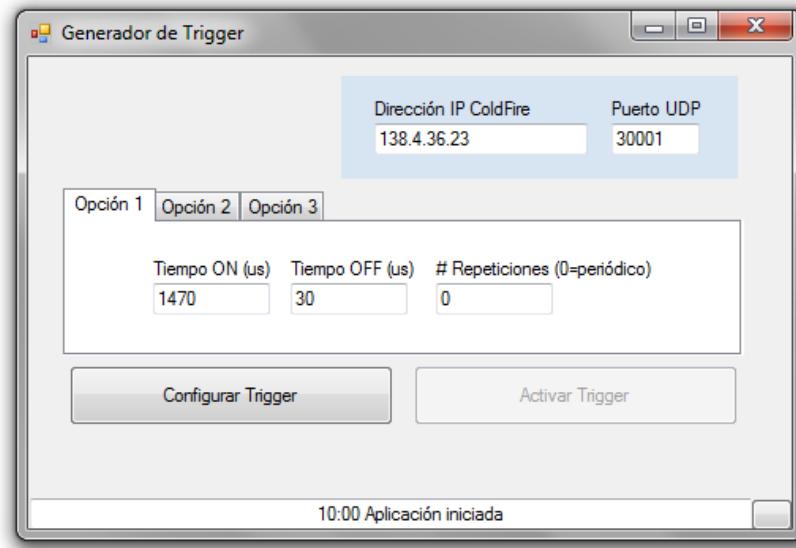


Figura C.9: Aplicación para configurar el *trigger* digital.

pestaña es similar a la segunda. Lo único que varia respecto de la anterior es que en lugar de definirse el periodo lo que se introduce como dato es la frecuencia de la señal.

La *GUI* permite definir la dirección IP del controlador y su puerto UDP. Incorpora un panel de mensajes inferior para poder mostrar la respuesta proporcionada por el microcontrolador.

La *GUI* ha sido finalmente implementada en la aplicación del sistema (véase 5.2) para integrar todo el software en una única aplicación.

Apéndice D

Documentación tarjeta digitalizadora.

D.1. Modos de disparo soportados.

Al margen de cual sea la fuente que provoque el inicio de la captura, el *ADC* permite configurar 4 modos distintos de disparo denominados *post-trigger*, *pre-trigger*, *middle-trigger* y *delay-trigger*. A continuación se pasa a detallar cada uno de ellos.

D.1.1. Post-Trigger.

En el modo *post-trigger* las muestras se comienzan a registrar en la memoria a partir de la señal de *trigger* según muestra la figura D.1

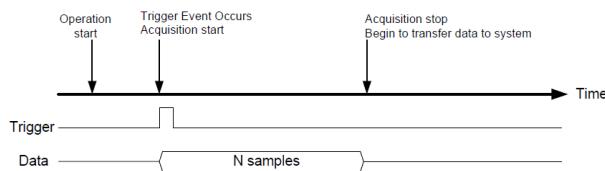


Figura D.1: Modo de captura *post-trigger*.

D.1.2. Pre-Trigger.

En el modo *pre-trigger* las muestras se van registrando en la memoria. Cuando se detecta el *trigger* el controlador transfiere las últimas N muestras registradas según se muestra en la figura D.2. Si en el momento de la captura no había suficientes muestras capturadas se espera a una nueva señal de *trigger*

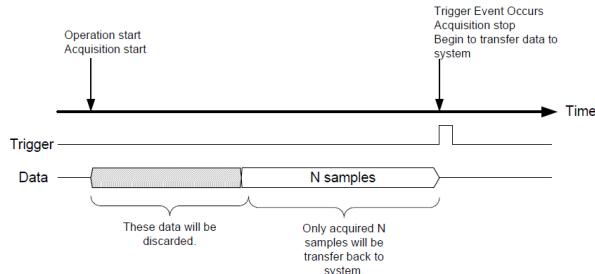


Figura D.2: Modo de captura *pre-trigger*.

D.1.3. Middle-Trigger.

El modo *middle-trigger* permite una combinación de los dos primeros modos según se muestra en la figura D.3. Las dimensiones de M y N son configurables por software.

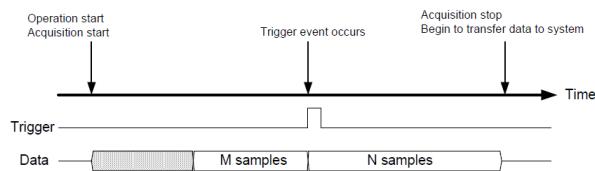


Figura D.3: Modo de captura *middle-trigger*.

D.1.4. Delay-Trigger.

Por último, en el modo de captura *delay-trigger* la captura comienza cuando transcurre un número programable de ciclos de reloj según se indica en la figura D.4.

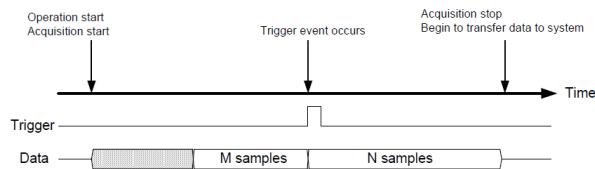


Figura D.4: Modo de captura *delay-trigger*.

D.2. Submuestreo.

El *ADC* incorpora un contador de 24 bits denominado *scan interval* y diseñado para decrementar la frecuencia de la fuente de reloj o *TIMEBASE*. De tal forma, la frecuencia de muestreo real se rige según la ecuación D.1

$$f_{muestreo} = \frac{TIMEBASE}{scan\ interval} \quad (D.1)$$

El submuestreo configurado es único y se aplica de igual modo a los 4 canales. En la figura D.5 se muestran las capturas realizadas en una configuración *post-trigger* para diferentes valores de *scan interval*.

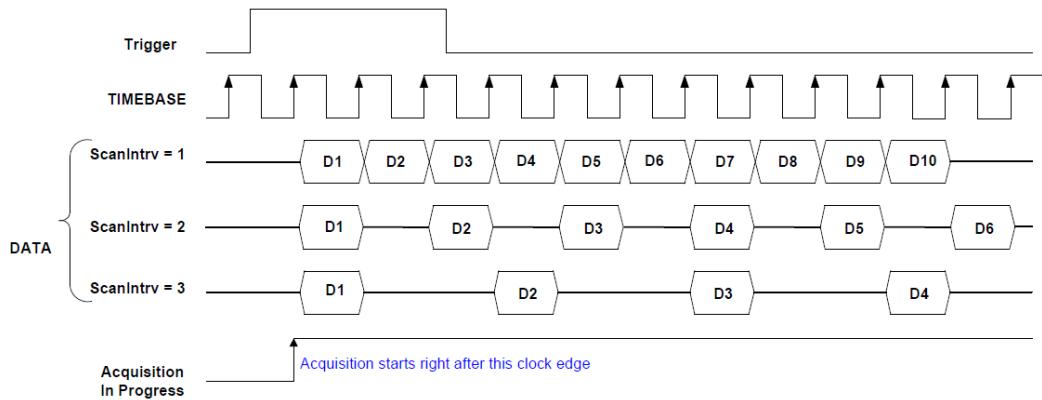


Figura D.5: Submuestreo aplicado según el valor de scan interval.

D.3. Modo de captura simple.

En el modo de captura simple, o *snapshot*, las muestras digitalizadas se almacenan en la memoria interna. Una vez finalizada la captura se inicia una transferencia DMA de los datos capturados. Este modo de trabajo esta limitado por el dimensionamiento de la memoria interna la cual permite únicamente almacenar un total de 256 Megamuestras.

La memoria disponible se reparte por igual entre los canales seleccionados para la captura. Así, por ejemplo se pueden capturar hasta 64 Megamuestras por cada uno de los 4 canales, o bien capturar 256 Megamuestras de un único canal.

D.4. Modo de captura continuo.

En el modo de captura continuo las muestras capturadas se almacenan también de forma previa en la memoria interna de 512MB (aplicable solo para versiones del driver *WD-DASK 1.1* y posteriores. Véase D.5). La escritura en la memoria se realiza siguiendo el patrón de una tabla circular tipo *FIFO*.

El *ADC* comienza al inicio de la captura a registrar datos recorriendo direcciones de memoria contiguas del primer banco. Cuando éste se completa, el *ADC* comienza

la escritura en el segundo describiendo de esta manera un proceso alterno y secuencial. Mientras el *ADC* registra las muestras capturadas en este otro segundo banco se transmiten los datos del primer banco mediante una transferencia DMA.

Para poder soportar este modo de trabajo y evitar pérdida de muestras es imprescindible que la transferencia DMA sea más rápida que el llenado de datos por parte del *ADC* ya que cada banco de memoria solamente puede ser direccionado por un único dispositivo. En el instante en el que el *ADC* comienza a registrar datos en uno de los bancos de memoria ningún otro dispositivo puede acceder a la lectura del mismo. Solamente se puede acceder a la lectura de éste cuando el *ADC* ha terminado de llenarlo y comienza a escribir en el banco alternativo. De tal forma, los datos que no se hayan transferido antes del comienzo del proceso de escritura serán sobrescritos.

En un bus PCI cualquier dispositivo que implemente el controlador *bus mastering* (como ocurre con el controlador incorporado en el *ADC*) puede tomar control del bus e iniciar una transferencia DMA. Para ello, el controlador dispone de un contador denominado *latency timer*. Dicho contador se decrementa por cada ciclo de reloj del bus.

Cuando el contador vale 0, el controlador detiene la transferencia y libera el bus. Si pasado un tiempo de guarda ningún otro controlador *bus mastering* ha tomado posesión del bus el controlador vuelve a tomar control del mismo y continua la transferencia.

Si los dispositivos conectados a un bus PCI tienen un temporizador demasiado corto se conmutará con demasiada frecuencia de dispositivo, lo que mermará la eficacia de transferencia del bus. Si, por el contrario, los dispositivos tienen temporizadores demasiado lentos se corre el riesgo de que no se atienda a tiempo a algún dispositivo y se pierda información o ésta se transfiera demasiado tarde.

El temporizador de latencia o *latency timer* es un parámetro ajustable en cada controlador *bus mastering*. No obstante, los drivers que facilita el fabricante utilizan el modelo *KMDF (Kernel-Mode Driver Framework)* para interactuar con el sistema operativo Windows. Dicho modelo dispone de una función denominada *WdfDmaEnablerGetMaximumLength* en la que el sistema operativo retorna al driver del dispositivo el máximo tamaño permitido en una transferencia DMA simple (véase [http://msdn.microsoft.com/es-es/library/windows/hardware/ff546986\(v=vs.85\).aspx](http://msdn.microsoft.com/es-es/library/windows/hardware/ff546986(v=vs.85).aspx))

En la arquitectura sobre la que se ejecuta actualmente la *aplicación del sistema* (véase 5.2) dicha función retorna un valor de 6MB. El driver consecuentemente configura el temporizador de latencia del controlador DMA de la tarjeta *ADC* para que se agote al transferir 6MB.

Una vez configurada e iniciada la captura en este modo, el *thread* encargado de la lectura de los datos (véase 5.2) ejecuta un bucle que llama repetidamente a la función *WD_AI_AsyncDblBufferHalfReady*. Dicha función retorna un valor *true* cuando se ha finalizado el registro de muestras en uno de los bancos y el controlador *ADC* está preparado para transferirlo.

Cuando *WD_AI_AsyncDblBufferHalfReady* retorna dicho valor *true* la aplicación llama a la función *WD_AI_DMA_Transfer*. La función comprueba que el tamaño de las

muestras a transferir desde la SDRAM del *ADC* tiene un tamaño o igual que la memoria reservada para su transferencia. En el caso de que la memoria reservada no tenga una tamaño suficientemente grande la aplicación realiza una llamada a la función *WD_ALDMA_TransferBySize* que divide la transacción DMA en varias trasferencias DMA simples del tamaño de la memoria reservada.

En las pruebas realizadas hasta la fecha se han podido realizar capturas en este modo de trabajo de 2 canales muestreados a 40 MHz sin pérdida de muestras hasta 540 MS.

D.5. Driver de la tarjeta.

El fabricante pone a disposición en su web (<http://www.adlinktech.com>) el software *DAQ-PILOT* (figura D.6)que permite un manejo simple de la tarjeta capturadora.



Figura D.6: Software *DAQ-PILOT*.

Para tener un mayor control sobre las operaciones de captura y transferencia es necesario utilizar la API WD-DASK de 32 bits que el fabricante también facilita en su misma página web. Dicha API se comunica con el driver cuya arquitectura es de 64 bits según se indica en la figura D.7.

La API esta soportada para los lenguajes de programación Visual Basic, Linux C/C++, Borland C++ y Visual C/C++. En lo sucesivo se hará referencia exclusivamente a este último lenguaje de programación.

A continuación se resumen las principales funciones de llamada de la API utilizadas en la *aplicación del sistema* (véase 5.2).

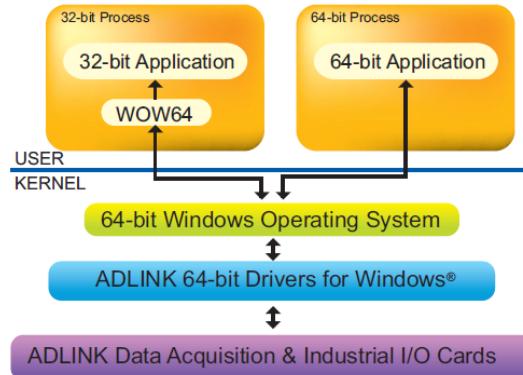


Figura D.7: Tarjeta ADC.

D.5.1. *WD_Register_Card.*

La función *WD_Register_Card* inicializa el hardware y retornando un *ID* que identifica a la tarjeta inicializada. Esta función debe de ser llamada en primer lugar antes de que otra función sea llamada ya que el resto de funciones utilizan como parámetro dicha *ID* de la tarjeta.

D.5.2. *WD_AI_ContBufferSetup.*

La función *WD_AI_ContBufferSetup* indica al controlador PCI la ubicación de los buffers donde se han de transferir las muestras capturadas en el modo de captura continuo. La primera llamada a dicha función determina la zona de la memoria destino para los datos registrados en el primer banco. Una segunda llamada consecutiva a dicha función configura la memoria destino para el segundo banco de memoria.

D.5.3. *WD_AI_CH_ChangeParam.*

La función *WD_AI_CH_ChangeParam* permite definir individualmente para cada canal el rango de entrada de la señal a capturar (± 0.2 V / ± 1 V), la impedancia de entrada ($1M\Omega$ / 50Ω) y el posible acople de continua.

D.5.4. *WD_AI_Config.*

La función *WD_AI_Config* define la fuente de reloj a utilizar durante la captura. Las posibles opciones son una fuente de reloj externa, una referencia interna de 40 MHz, y una señal recibida a través del bus SSI.

D.5.5. ***WD_AI_Trig_Config.***

La función *WD_AI_Trig_Config* se utiliza para definir la fuente de *trigger* (externo, software, analógico, bus SSI) y el modo de *trigger* (véase la sección D.1). Las opciones *pre-trigger* y *middle-trigger* no se permiten configurar para el modo de captura continuo (véase la sección D.4).

D.5.6. ***WD_AI_Trig_Config.***

La función *WD_AI_AsyncDblBufferMode* activa o desactiva el modo de captura continuo (véase la sección D.4).

D.5.7. ***WD_AI_ContScanChannels.***

La función *WD_AI_ContScanChannels* determina el número de muestras a capturar para el caso de modo de captura simple (véase D.3) y el valor de la variable *scan interval* para configurar un posible submuestreo (véase la sección D.2).

D.5.8. ***WD_AI_AsyncDblBufferHalfReady.***

La función *WD_AI_AsyncDblBufferHalfReady* retorna un valor *true* cuando el controlador *ADC* ha terminado de transferir uno de los dos bancos de memoria al buffer de la aplicación.

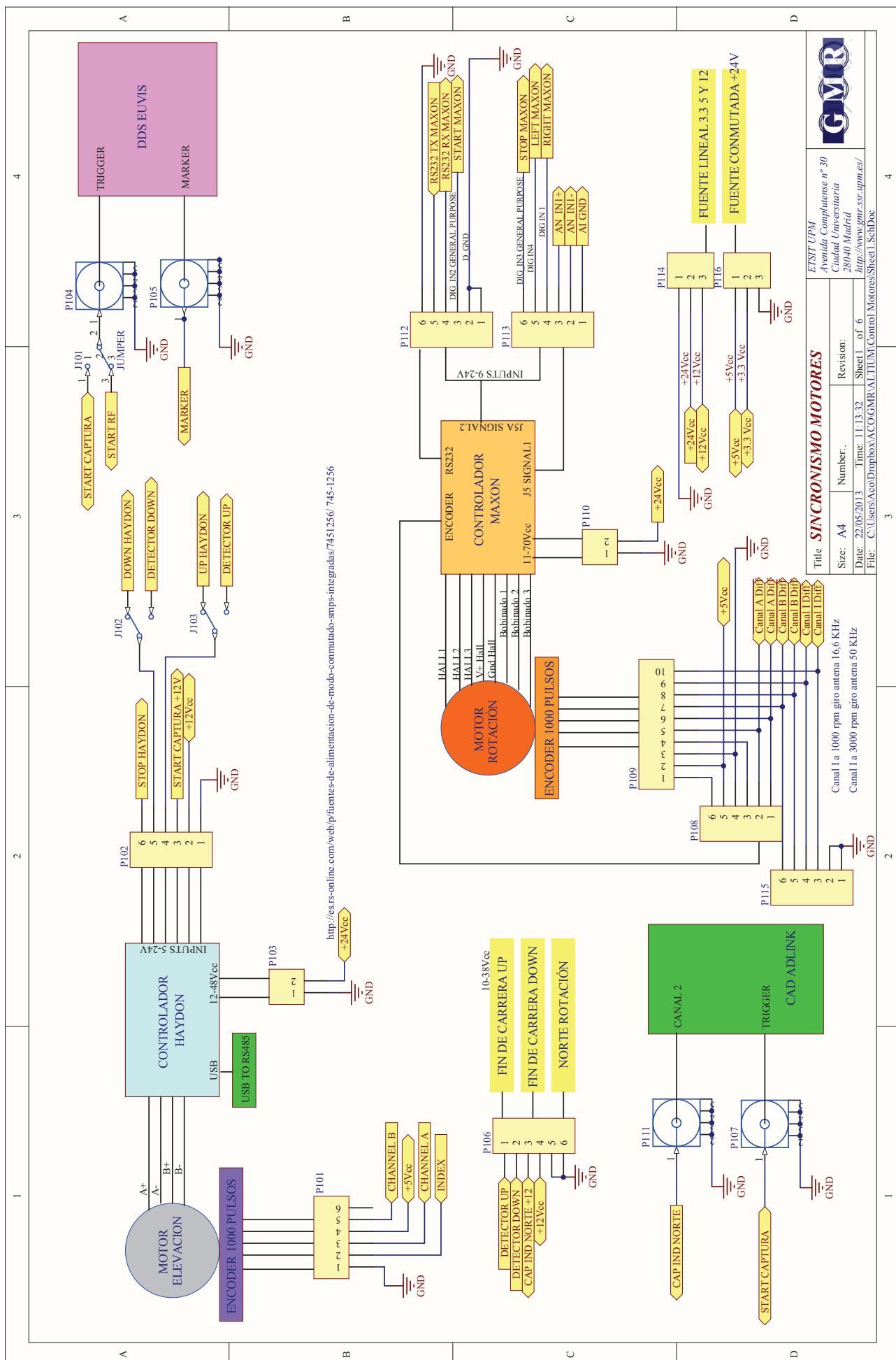
D.5.9. ***WD_AI_AsyncClear.***

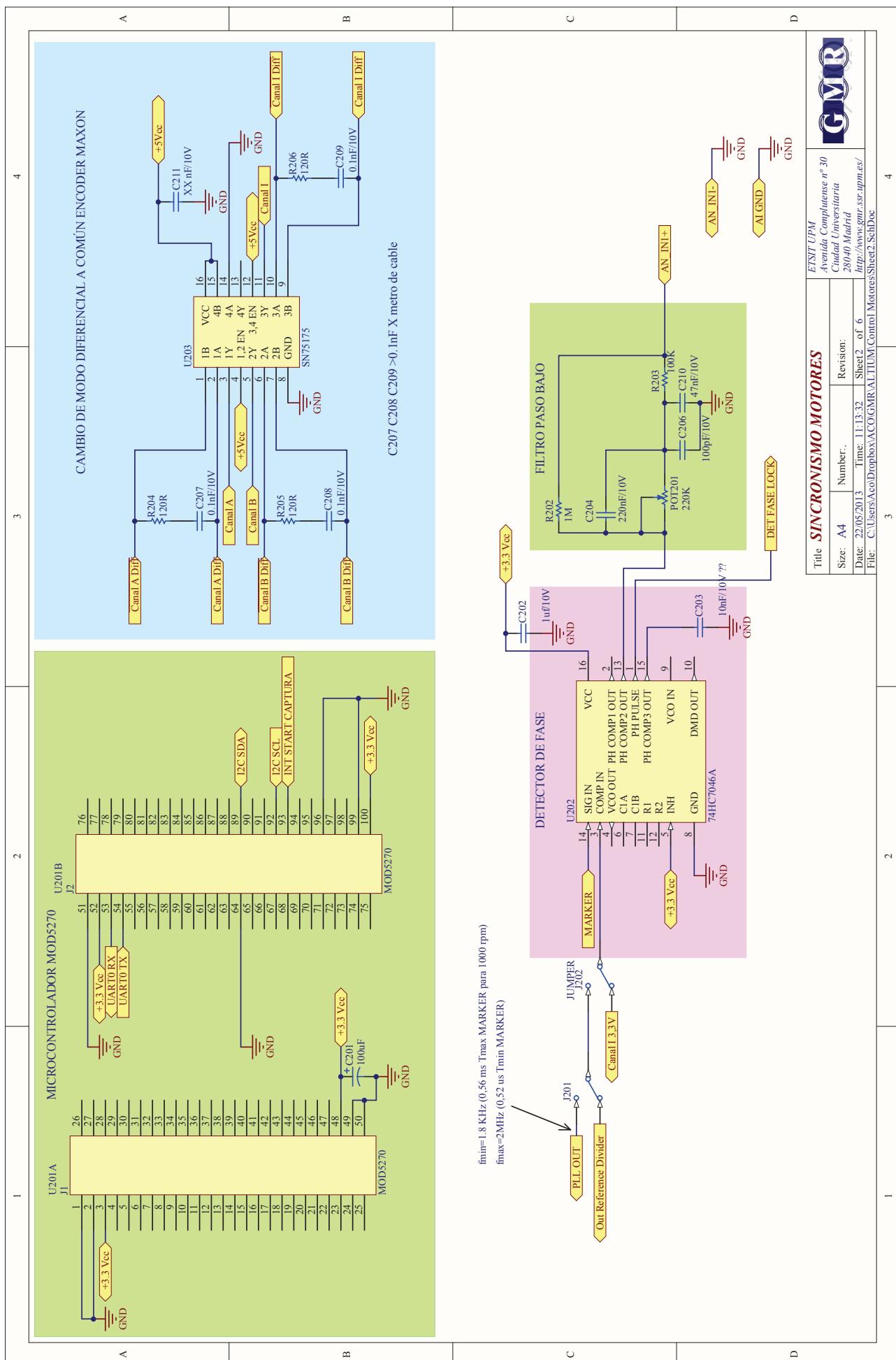
La función *WD_AI_AsyncClear* indica al *ADC* que ha de finalizar con el registro de muestras. Dicha función sirve para cancelar una captura simple o para finalizar el modo de captura continuo.

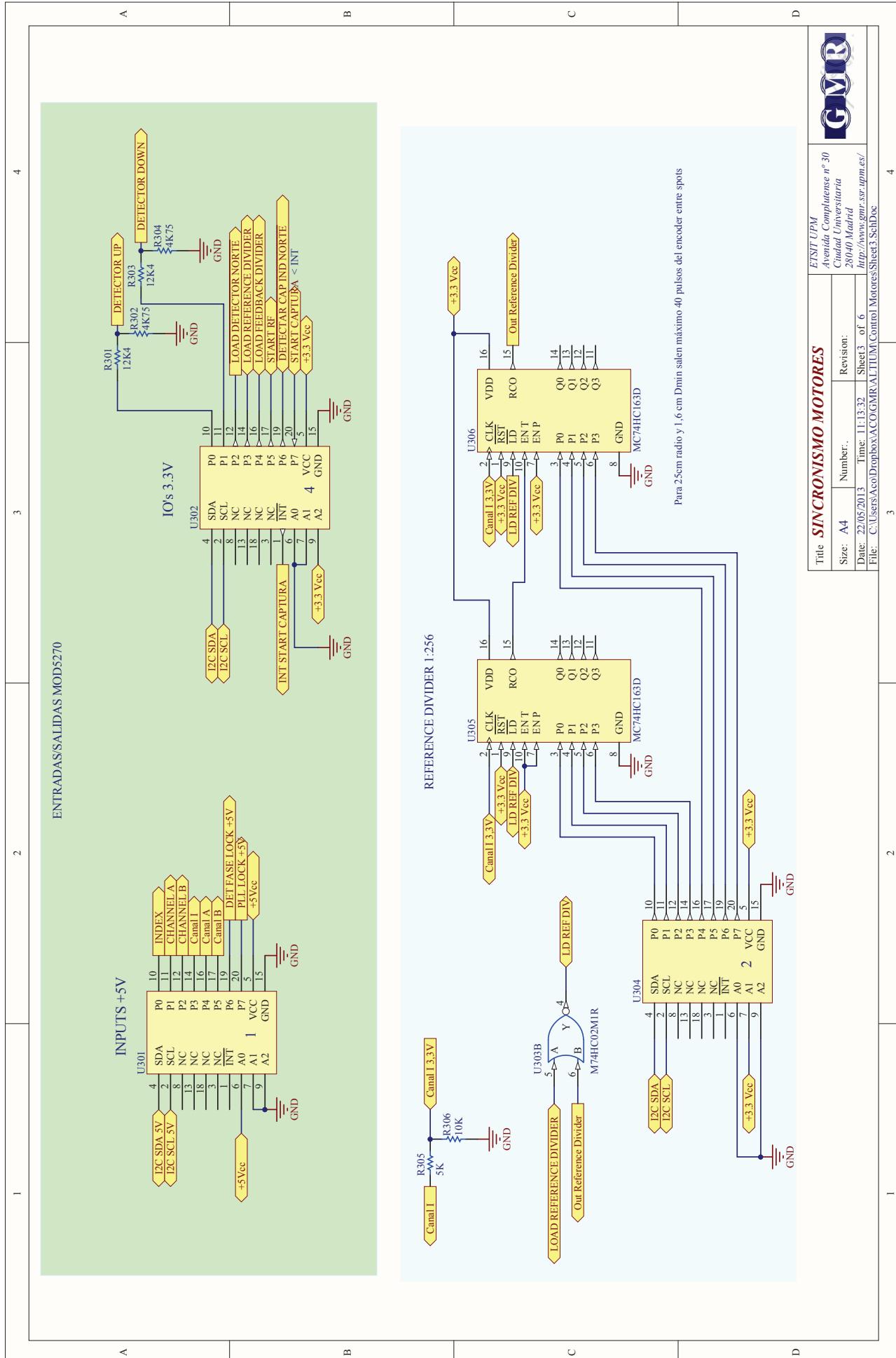
Apéndice E

Documentación tarjeta de inicio de captura.

E.1. Esquemas tarjeta de inicio de captura.







GMAR

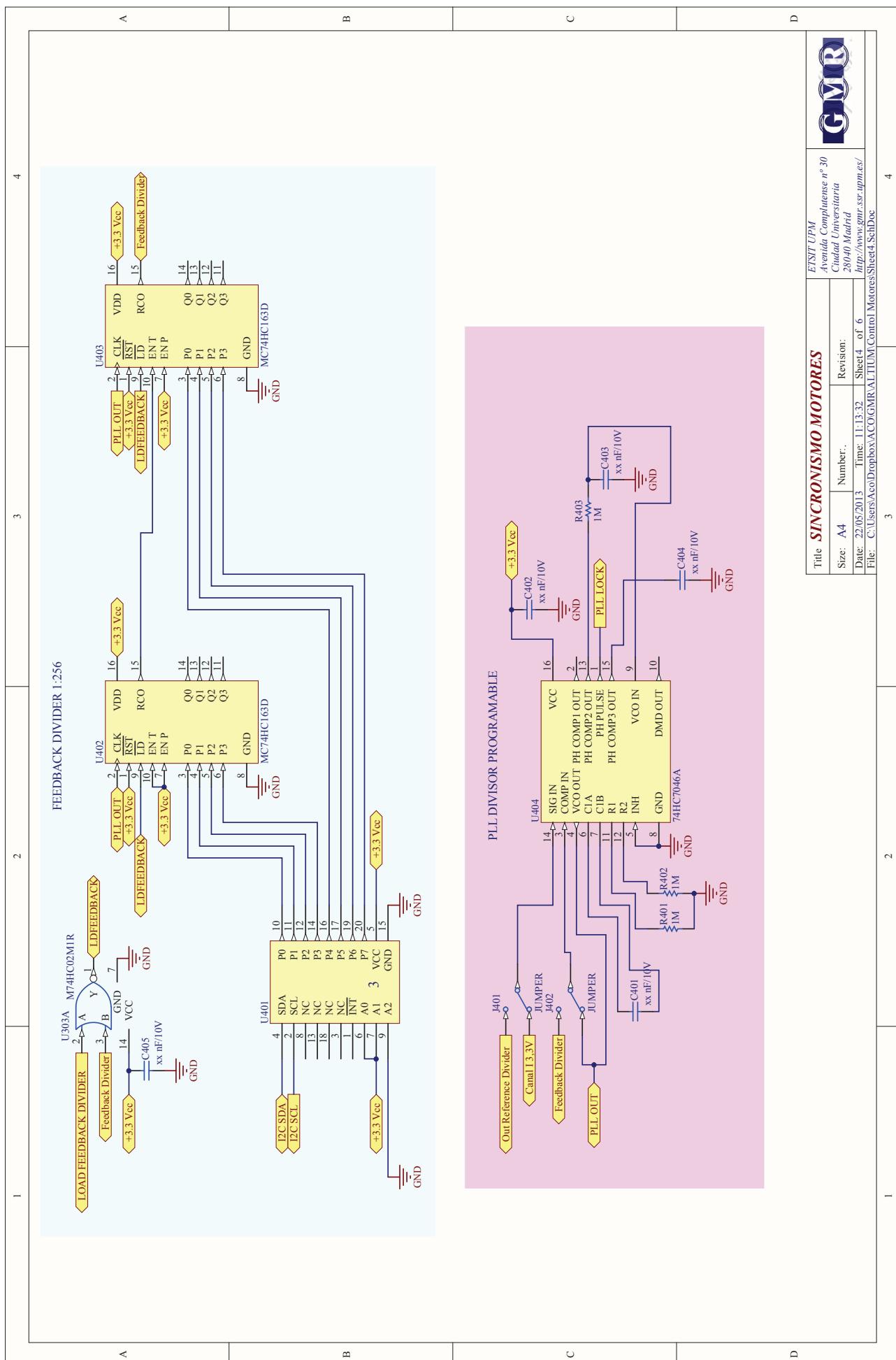
TITLE SÍNCRONISMO MOTORES

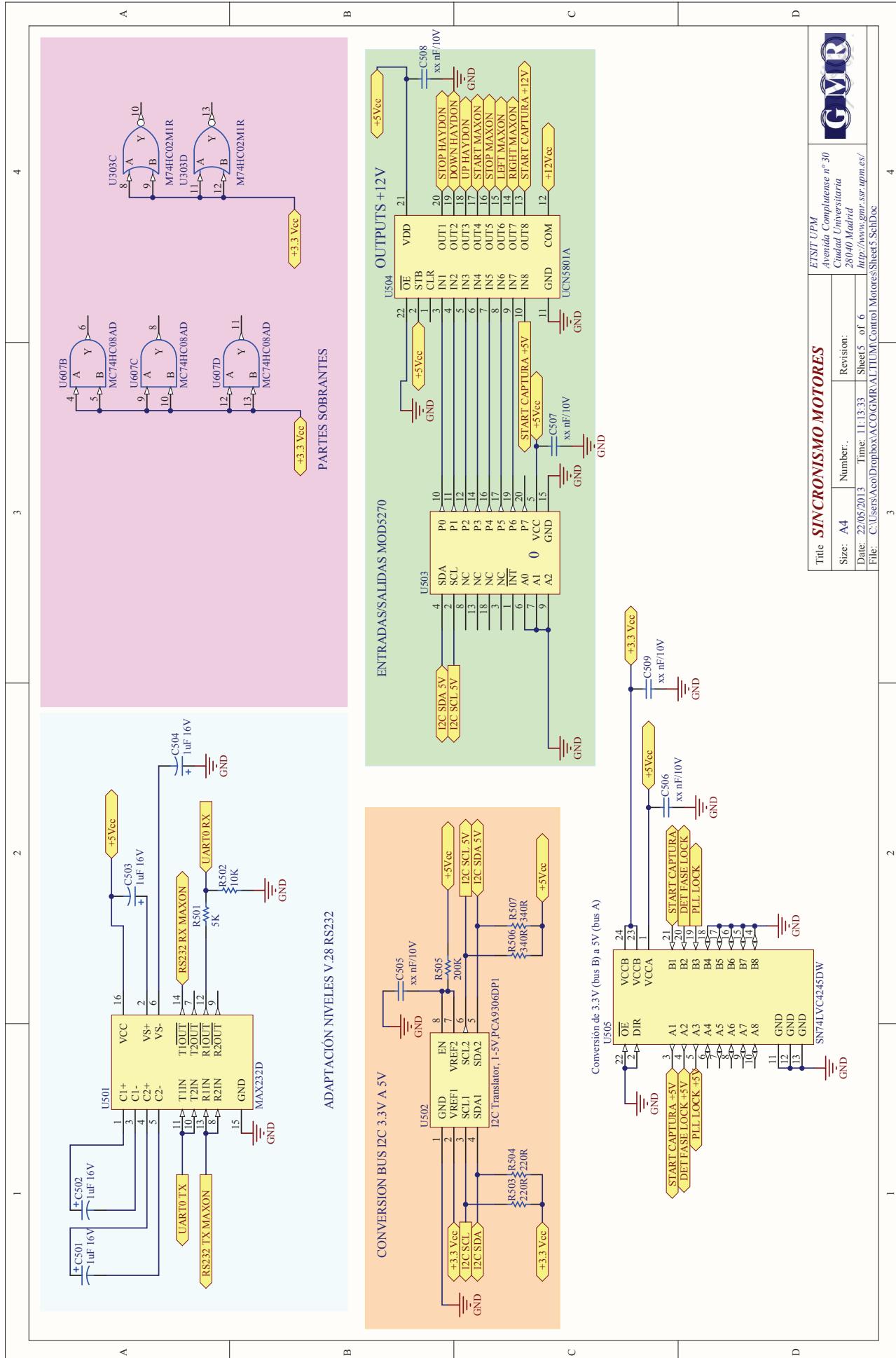
ETSII UPM

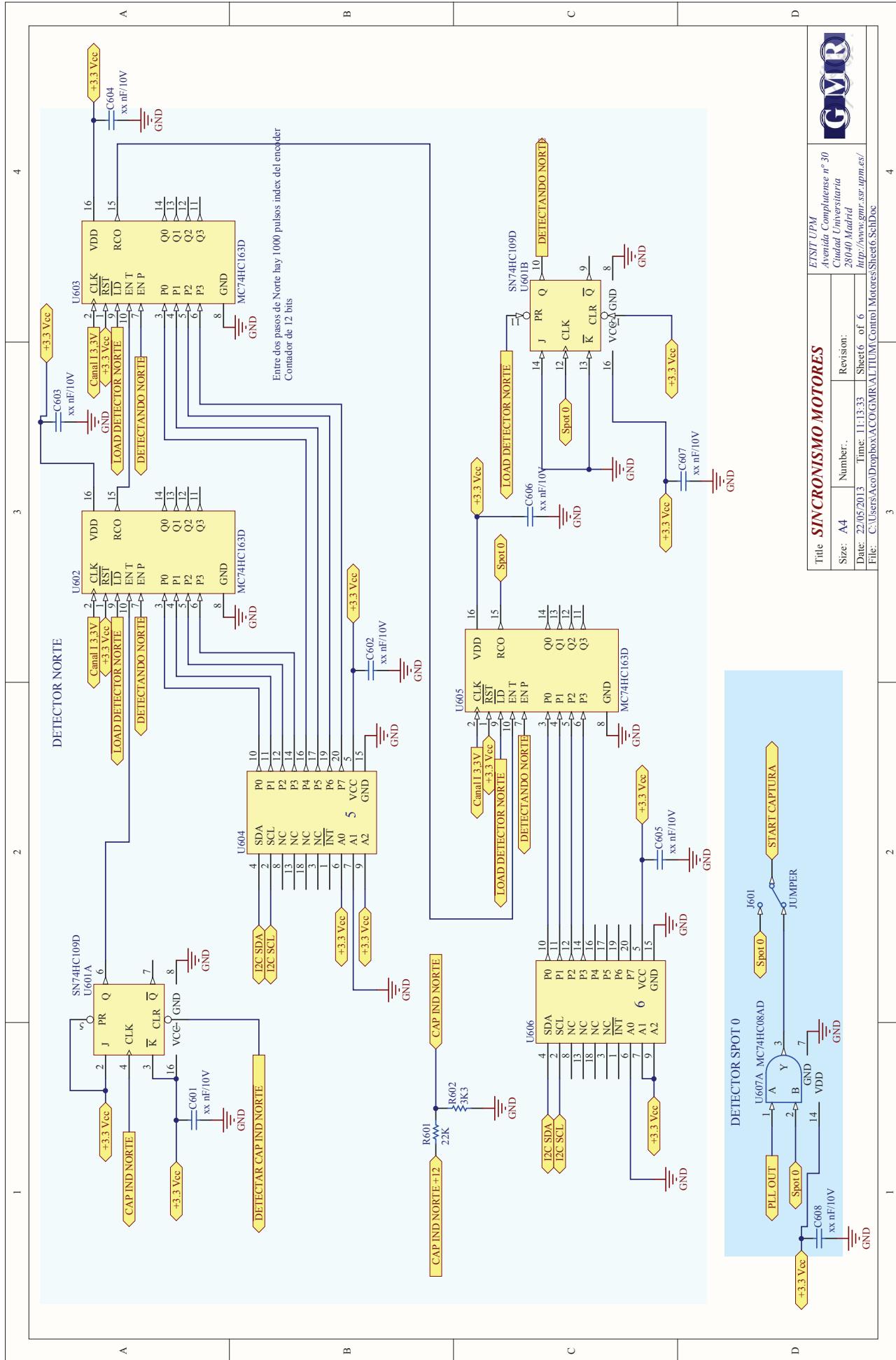
Avenida Complutense nº 30
Ciudad Universitaria
28040 Madrid

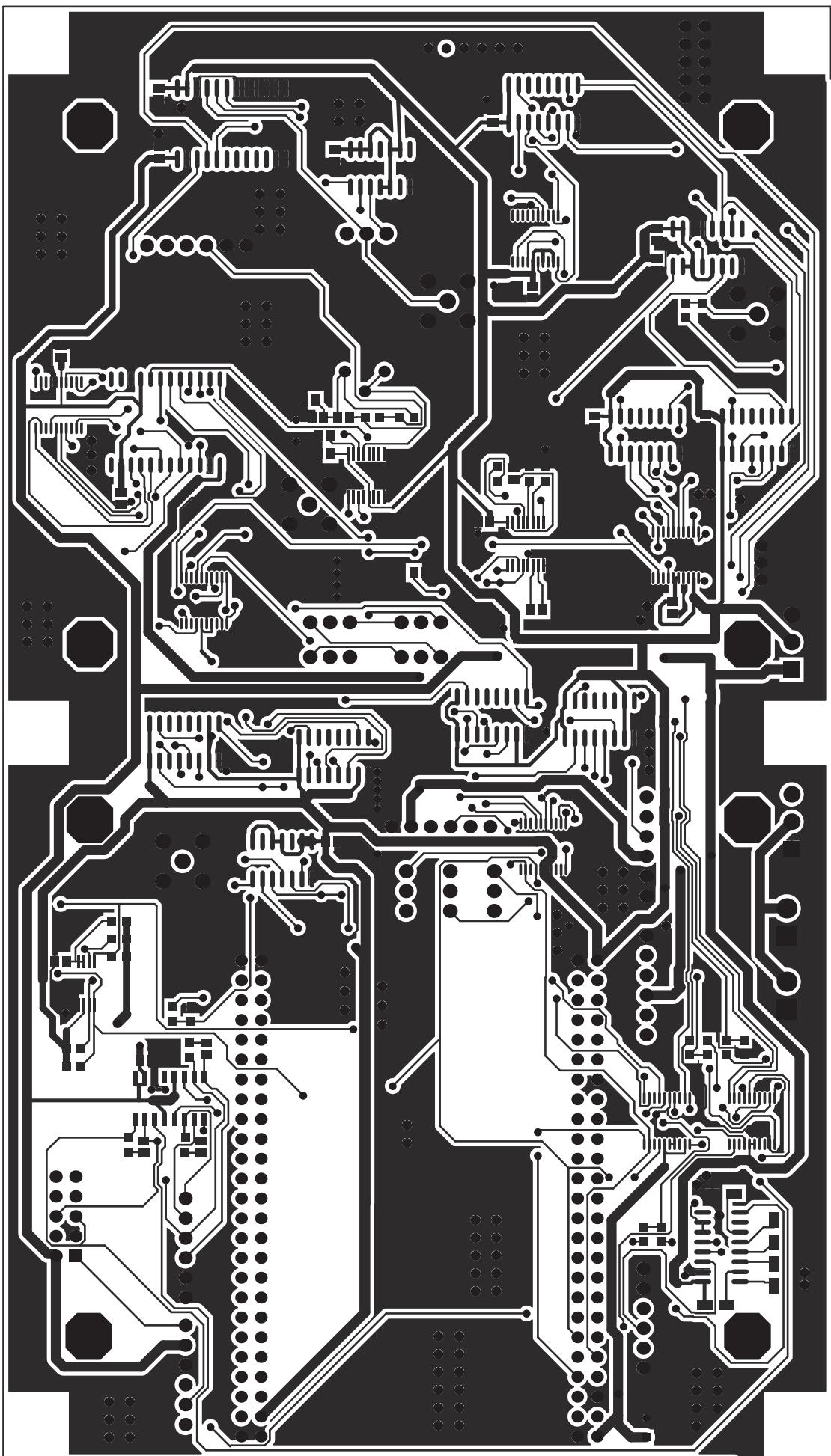
<http://www.grc.ssr.upm.es/>

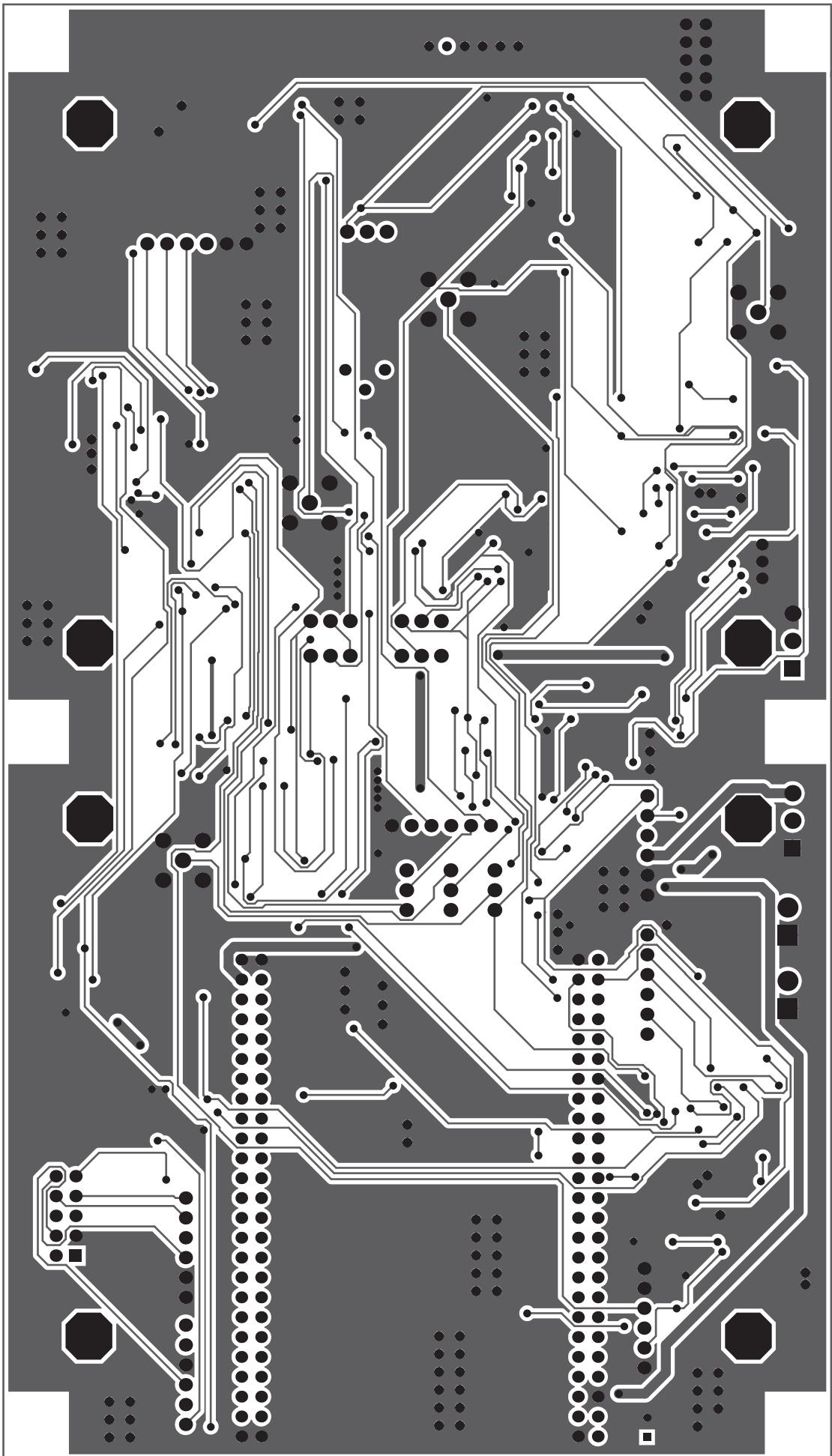
Sheet 3 of 6
File: C:\Users\ACO\Dropbox\ACOGMRLITUM\Control Motores\Sheet3\SchDoc

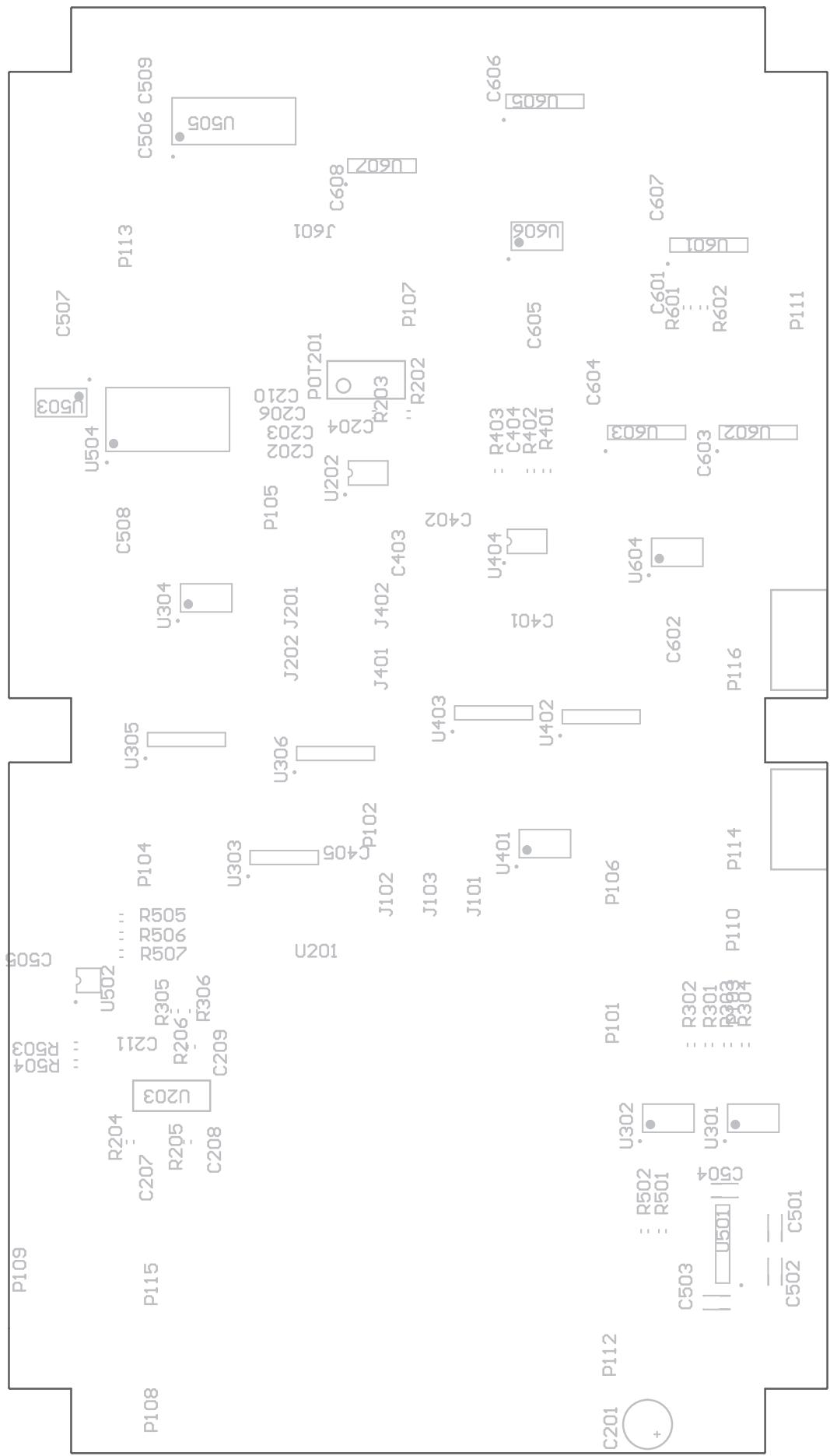


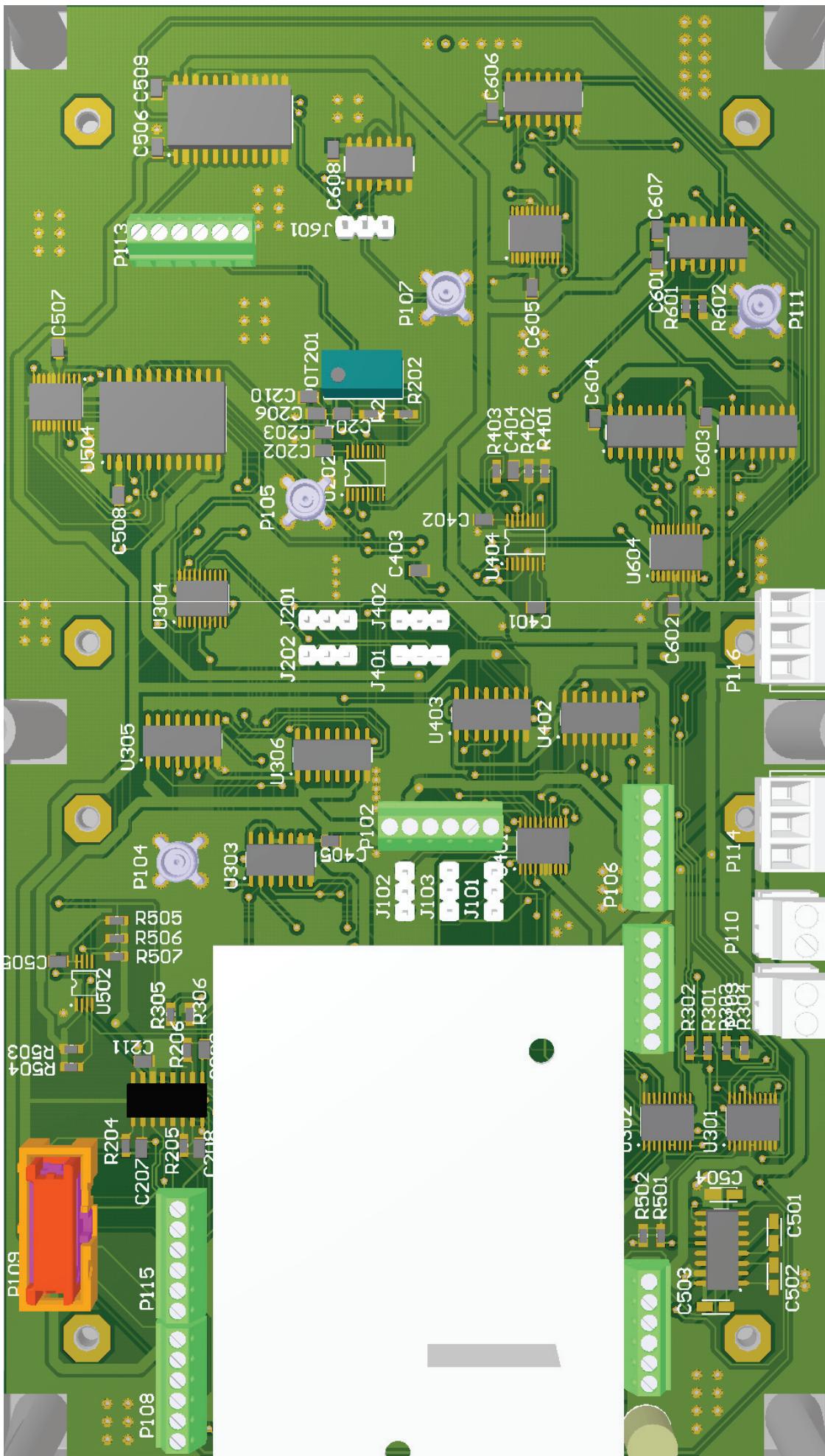












E.2. Código tarjeta de inicio de captura.

main.cpp

```
/****************************************************************************
 * Programa para programar contadores de la tarjeta de inicio de
 * captura. Alejandro Badolato 07/03/2012
 **************************************************************************/
#include <NetworkDebug.h>
#include "predef.h"
#include <stdio.h>
#include <ctype.h>
#include <startnet.h>
#include <autoupdate.h>
#include <..\MOD5270\system\sim5270.h>
#include <cfinter.h>
#include <utils.h>
#include <pins.h>
#include "i2cmaster.h"
#include <string.h>
#include <taskmon.h>
#include <smarttrap.h>
#include <udp.h>
#include <stdlib.h>
#include <serial.h> // para gestión UARTs

#define u503 0x38 // Tranceptor's addresses
#define u301 0x39
#define u304 0x3A
#define u401 0x3B
#define u302 0x3C
#define u604 0x3D
#define u606 0x3E
#define transceptorWriting 0xFE // Tranceptor's bus writing bit1=0
#define ipDestino "138.4.36.21"
#define puertoDestino 30001

extern "C"{
void UserMain( void *pd );
void SetIntc( long func, int vector, int level, int prio );
void ModificaPITR( int pitr_ch );
void envioUDP( char * buf );
void I2CEscribe( BYTE address, BYTE *dato, BYTE mascara );
BYTE I2CLee( BYTE direccion );
void UdpReaderMain( void * pd );
void trataMensaje( int funcion, BYTE *valor, BYTE mascara );
BYTE testTransecptor ( BYTE address, BYTE mascara );
BYTE Ascii2Byte( char* buf );
void testTransceptores( void );
void trataMensajeUdp( char* buf );
void setDetectorNorte( BYTE *valor );
void leeru301( void );
void setReferenceDivider( BYTE *valor );
void setFeedBackDivider( BYTE *valor );
void RXRS232 (int fd);

}

const char *AppName = "Control Motores"; // App name for IPSetup
char mensaje[ 80 ] = "57";
BYTE buffer[ I2C_MAX_BUF_SIZE ];
char I2CInputBuffer[ I2C_MAX_BUF_SIZE ]; // User created I2C input buffer
char* inbuf = I2CInputBuffer; // Pointer to user I2C buffer
BYTE address = 0;
BYTE I2CStat;
```

```

main.cpp

int fd0=0; // descriptor de ficheros para RS232

// Allocate task stack for UDP listen task
DWORD UdpTestStk[ USER_TASK_STK_SIZE ];

///////////////////////////////
// UserMain
//
void UserMain( void *pd ){
    int portnum = 30001;
    InitializeStack();
    // inicializar debub por red cuando sea necesario
    #ifdef _DEBUG
        InitializeNetworkGDB();
    #endif // _DEBUG
    OSChangePrio( MAIN_PRIO );
    EnableAutoUpdate();
    EnableTaskMonitor();
    iprintf( "Aplicacion iniciada\r\n" );
    I2CInit(0x1E); // máx 100KHz 1E= 25KHz fclock I2C
    OSTaskCreate( UdpReaderMain, (void *)portnum, &UdpTestStk[USER_TASK_STK_SIZE],
UdpTestStk,MAIN_PRIO - 1 );
    SerialClose( 0 );
    //Open the serial port: index=0x2002 subindex=0x00 dato 05=115200 baudrate
    // startbit=1 stopbit=1 databit=8 parity=none
    fd0 = OpenSerial( 0, 115200, 1, 8, eParityNone );
    // write( fd0, "10010320010288A8", 16 );
    write( fd0, "011030021020888A", 16 );
    while (1) {
        RXRS232( fd0 );
    }
}

// thread de recepción de mensajes UDP
void UdpReaderMain(void * pd){
    int port = (int)pd; // cast void * param as int port number
    char ack[80] = "Recibido mensaje.";
    char *buf;
    int len;
    iprintf("Listening on port: %d\r\n", port);
    // Create a FIFO for the UDP packet and initialize it
    OS_FIFO fifo;
    OSFifoInit( &fifo );
    // Register to listen for UDP packets on port number 'port'
    RegisterUDPFifo( port, &fifo );
    while (1){
        UDPPacket upkt( &fifo, 3 * TICKS_PER_SECOND );
        // Did we get a valid packet, or just time out?
        if ( upkt.Validate() ){
            // obtener pmr.on
            buf = (char*)upkt.GetDataBuffer();
            len = upkt.GetDataSize();
            buf[len] = '\0';
            iprintf( "Recibido: " );
            iprintf( (char*) buf );
            iprintf( "\n\r" );
            //envioUDP(ack); // envia por udp ACK
            trataMensajeUpd( buf );
        } //end if
    } // end while
} // end udpreadermain

```

main.cpp

```
// Transmite un BYTE por I2C
// lee primero del transceptor, aplica mascara para no alterar bits
// mascara = 0 x FF no pone a cero ningún bit
// address = dirección del transceptor
void I2CEscribe( BYTE address, BYTE *dato, BYTE mascara ){
    BYTE temp;
    temp = I2CLee( address );
    temp = temp & mascara;
    temp = temp | *dato;
    dato = & temp;
    // Send the buffer size plus one to include the null character
    I2CStat = I2CSendBuf( address, dato, 1 ); //strlen( (const BYTE*)dato)
    if ( I2CStat == I2C_OK )
        iprintf( "Sent successfully\r\n" );
    else
        iprintf( "Failed to send to address %x due to error: %d\r\n", address, I2CStat );
}

// Lee un Byte por I2C
BYTE I2CLee( BYTE address ){
    I2CStat = ( I2CReadBuf(address, buffer, 2) );
    if( I2CStat == I2C_OK ){
        iprintf( "Receive successfully\r\n" );
    } else{
        iprintf( "Failed to read due to error: %d\r\n", I2CStat );
    }
    return buffer[ 0 ];
}

// trata los mensajes recibidos
// función:
//      1=fijar valor contador detector norte
//      2=fijar valor reference divider
//      3=fijar valor feedback divider
//      4=fijar valor u503
//      5=fijar valor u302
//      6=fijar valor u504
//      7=leer valor u301
// máscara: máscara de escritura 0x00 escribe en todos los bits
void trataMensaje( int función, BYTE *valor, BYTE mascara ){
    switch ( función ){
        case 1:{ //fijar valor contador detector norte
            setDetectorNorte( valor );
            break;
        }
        case 2:{ //fijar valor reference divider
            setReferenceDivider( valor );
            break;
        }
        case 3:{ //fijar valor feedback divider
            setFeedBackDivider( valor );
            break;
        }
        case 4:{ //escribir en u503
            *valor = *valor | 0x01; // fallo en PCB aplicar STROBE siempre!!! (es in1)
            I2CEscribe( u503, valor, mascara );
            break;
        }
        case 5:{ //escribir en u302
            I2CEscribe( u302, valor, mascara );
    }
}
```

```

main.cpp

        break;
    }
    case 6:{ //escribir en u606
        I2CEscribe( u606, valor, mascara );
        break;
    }
    case 7:{ //leer valor u301
        leeru301();
        break;
    }
    case 8:{ //test Transceptores de BUS
        testTransceptores();
        break;
    }
    case 9:{ //test Transceptores de BUS
        write( fd0, "01", 2 );
        break;
    }
}
} // tratamensaje

// ajusta el valor del referenceDivider
void setReferenceDivider( BYTE *valor){
    BYTE *p;
    BYTE temp = 0x10;
    p= &temp;
    I2CEscribe( u304, valor, 0X00);
    I2CEscribe( u302, p, ~0x08 ); // load referenceDivider
    temp = 0x00;
    I2CEscribe( u302, p, ~0x08 ); // load referenceDivider
}

// ajusta el valor del feedBackDivider
void setFeedBackDivider( BYTE *valor ){
    BYTE *p;
    BYTE temp = 0x10;
    p = &temp;
    I2CEscribe( u401, valor, 0X00 );
    I2CEscribe( u302, p, ~0x10 ); // load feedbackDivider
    temp = 0x00;
    I2CEscribe( u302, p, ~0x10 ); // load feedbackDivider
}

// envía por UDP las entradas de u301
void leeru301( void ){
    char buf[ 2 ];
    char *f = buf;
    BYTE dato = I2CLee( u301 );
    itoa( (int)dato, f, 10 );
    envioUDP( f );
}

// chequea los expansores de bus
void testTransceptores( void ){
    BYTE test = 0x00; // tests OK
    char temp[ 10 ] = "Test OK";
    char temp2[ 12 ] = "Error Test";
    char *p;
    test = test | testTransecptor( u503, 0xFF ); // salidas +12V
    //test=test | testTransecptor(u301); // entradas
}

```

```

main.cpp

test = test | testTransecptor( u304, 0xFF ); // reference divider
test = test | testTransecptor( u401, 0xFF ); // feedback divider
test = test | testTransecptor( u302, 0x7C ); // p0,p1,p7 entradas resto salidas 3.3V
test = test | testTransecptor( u604, 0xFF ); // detector norte bits 1-8
test = test | testTransecptor( u606, 0xFF ); // detector norte bits 9-12
p = temp;
if (test) p = temp2;
envioUDP( p );
}

// chequea un transceptor de bus
// en aquellos bits donde la máscara = 1
BYTE testTransecptor ( BYTE address, BYTE mascara ){
    BYTE temp = 0x00; //0=test ok
    BYTE temp2 = 55 & mascara;
    I2CEscribe( address, &temp2, 0x00 );
    if ( temp2 != I2CLee( address ) ) temp = 0xFF;
    temp2 = 0xAA & mascara;
    I2CEscribe( address, &temp2, 0x00 );
    if ( temp2 != I2CLee( address ) ) temp = 0xFF;
    return temp;
}

// The following function converts the first two ascii chars of a string to a single hex value
BYTE Ascii2Byte( char* buf ){
    char conv[] = {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C', 'D', 'E', 'F'};
    BYTE temp = 0;
    for( int i = 0; i < 16; i++ )
        if ( toupper(buf[1]) == conv[i] )
            temp = i;
    for( int i = 0; i < 16; i++ )
        if ( toupper(buf[0]) == conv[i] )
            temp |= i << 4;
    return temp;
}

// estructura de los mensajes= funcion:valor:máscara
void trataMensajeUdp( char *buf ){
    int funcion = 0;
    BYTE valor[ 2 ] = { 0x00,0x00 };
    BYTE mascara = 0x00;
    char buffer2[ 8 ];
    char buffer3[ 8 ];
    char buffer4[ 16 ];
    char c = ':';
    char *c2, *c3, *c4, *c5;
    int len;
    c2 = strchr( buf, c ); // detectar string función
    len = c2 - buf;
    buffer2[ 0 ] = '\0';
    c3 = buffer2;
    strncpy( c3, buf, len );
    c3[ len ] = '\0';
    funcion = atoi( c3 );//leer función
    // detectar string valor
    c3 = strchr( c2 + 1, c );
    len = c3 - c2;
    buffer3[ 0 ] = '\0';
    c4 = buffer3;
    strncpy( c4, c2+1, len-1 );
    c4[ len - 1 ]='\'\0';
}

```

```

main.cpp

valor[ 0 ] = Ascii2Byte( c4 );
valor[ 1 ] = Ascii2Byte( c4 + 2 );
// detectar string máscara
len = strlen( buf );
len = buf + len - c3;
buffer4[ 0 ] = '\0';
c5 = buffer4;
strncpy( c5, c3+1, len-1 );
c5[ len - 1 ] = '\0';
mascara = Ascii2Byte( c5 );
trataMensaje( funcion, valor, mascara);
}

// envia mensaje por UDP
void envioUDP( char *f ){
    UDPPacket pkt;
    IPADDR ipaddr;
    //enviar me por udp
    pkt.SetSourcePort( puertoDestino ); //usamos el mismo para rx y tx
    pkt.SetDestinationPort( puertoDestino );
    pkt.AddData( f );
    pkt.AddDataByte( 0 );
    ipaddr = AsciiToIp( ipDestino );
    pkt.Send( ipaddr );
    sprintf( "Enviado: " );
    sprintf( (char*)f );
    sprintf( "\r\n");
}

// fija los contadores del detector de norte
// valor es un puntero a 2 bytes valor=MSB y valor+1=LSB
void setDetectorNorte( BYTE *valor ){
    BYTE *p;
    BYTE temp = 0x00;
    p= &temp;
    char mensaje[]="Fijado detector Norte\0";
    temp = 0x00;
    I2CEscribe( u302, p, ~0x40 ); // detectar captador inducción norte= low= desactiva
counters
    // escribimos en transceptores valores para carga en p0 p3
    I2CEscribe( u604, valor + 1, 0X00 );// valor+1=LSB
    I2CEscribe( u606, valor, 0X00 );// valor=MSB
    temp = 0x00;
    I2CEscribe( u302, p, ~0x04 ); // load detector norte = low =load
    OSTimeDly(1); // load serie. Necesitamos un pulso de reloj por lo menos 50 ms demoramos la
carga
    temp = 0x04;
    I2CEscribe( u302, p, ~0x04 ); // load detector norte =high (desactivado)
    // carga realizada
    temp = 0x40;
    I2CEscribe( u302, p, ~0x40 ); // detectar captador inducción norte = high = activa
contadores en prox paso por norte
    envioUDP(mensaje);
}

void RXRS232 (int fd){
    //Set up a file set so we can select on the serial ports...
    fd_set read_fds;
    FD_ZERO( &read_fds );
    FD_SET( fd, &read_fds );
    if ( select( FD_SETSIZE, &read_fds, (fd_set *) 0, (fd_set *) 0, TICKS_PER_SECOND * 5
) )

```

```
main.cpp

{
    if ( FD_ISSET( fd, &read_fds ) )
    {
        char buffer[40];
        int n = read( fd, buffer, 40 );
        write( fd, buffer, n );
        // 1030021020888A resto de bytes a enviar
    }
}
else
{
    //timeout...
}
}
```


Apéndice F

Código Matlab.

F.1. Código para la determinación de las coordenadas del blanco.

```

%% función que devuelve una matriz con las coordenadas de los blancos
%% parámetros entrada:
%%     velrotacion (rpm) velocidad de giro del motor maxon
%%     velBarrido (steps por segundo) velocidad motor haydon
%%     versionMotorBarrido (B o C) versión del motor haydon
%%     mustreo (muestras por segundo) muestreo del ADC
%%     muestrasConDeteccion (array) [#muestra_deteccion1,#muestra_deteccion2...]
%% parámetros salida:
%%     x (m) array con coordenadas x [detección1, detección2....] en eje Xa
%%     y (m) array con coordenadas y [detección1, detección2....] en eje Ya con origen ↴
%% en el suelo

function [x,y]=coordenadasBlanco(velRotacion,velBarrido,versionMotorBarrido,mustreo,↙
muestrasConDeteccion)
%%%%%%%%%%%%%
% definición de parámetros iniciales
%%%%%%%%%%%%%
a0 = -0.02586; %coeficientes del polinomio
a1 = -3.08996;
a2 = 2.89638;
a3 = 0.43756;
a4 = -6.60288;
a5 = -3.85521;
b1 = -2.51789;
b2 = 1.77152;
b3 = -3.76793;
b4 = -2.07521;
beta=4.557198; % ángulo de inclinación eje del espejo respecto eje rotación (grados)
gammaMin=-6.8;
if versionMotorBarrido == 'B'
    avanceMmPorStep=0.0158; % versión B del motor
else
    avanceMmPorStep=0.0317; % versión C del motor . mm de avance por step
end
gradosPorMmExtension=0.5/1.12; % 0,5° por cada 1.12mm de extensión -6.8 a -1.5
gradosPorMmExtension2=0.5/1.13; % de -1.5° a 3°
gradosPorMmExtension3=0.5/1.135; % de 3° a 6.8°
rpmGiroPhi=velRotacion/4;
gradosPorSegPhi=rpmGiroPhi*6;
tiemposDeteccion=muestrasConDeteccion*(1/mustreo);
phi=gradosPorSegPhi*tiemposDeteccion;
phi=mod(phi,360);
steps=[1:1:1942]; % con 1942 se llega hasta +6.8 con la versión B
gamma=steps*avanceMmPorStep*gradosPorMmExtension;
temp1=find((gamma+gammaMin)>-1.5);
stepsTramo1=steps(temp1(1)-1); % total steps en final tramo1
faltanTramo2=(steps-stepsTramo1); % resto de steps que faltan (del tramo2)
gamma(temp1(1):end)=((stepsTramo1*gradosPorMmExtension)+(faltanTramo2(temp1(1):end) ↴
*gradosPorMmExtension2))*avanceMmPorStep ; % grados motor elevación
temp2=find((gamma+gammaMin)>3);
stepsTramo2=steps(temp2(1)-1); % total steps en final tramo2
faltanTramo3=(steps-stepsTramo2); % resto de steps que faltan (del tramo2)
gamma(temp2(1):end)=((stepsTramo1*gradosPorMmExtension)+((stepsTramo2-stepsTramo1) ↴
*gradosPorMmExtension2)+faltanTramo3(temp2(1):end)*gradosPorMmExtension3) ↴

```

```
*avanceMmPorStep ; % grados motor elevación
gamma=gamma + gammaMin;
stepsParaSpot=velBarrido*tiemposDeteccion;
stepsParaSpot=floor(stepsParaSpot);
gamma=gamma(stepsParaSpot+1);
%%%%%%%%%%%%%
%   conversión de unidades
%%%%%%%%%%%%%
beta=beta*pi/180; % radianes
gamma=gamma*pi/180; % radianes
phi=phi*pi/180; % radianes

%%%%%%%%%%%%%
% cálculo coordenadas en el sistema de
% la antena Xa e Ya
%%%%%%%%%%%%%
u=-sin(beta)*cos(phi);
v=sin(gamma)*cos(beta) + (cos(gamma)*sin(beta)).*sin(phi);
x= a0 + a1*u + a2*u.^2 + a3*v.^2 + a4*u.^3 + a5*u.*v.^2;
y= b1*v + b2*u.*v + b3*u.^2.*v + b4*v.^3;
y=y+1.147; % añadimos offset para medida desde el suelo
```

F.2. Código para representar gráficamente los blancos.

```
clear all;
close all;
load 'deteccionesAlejandro.mat'
[x,y]=coordenadasBlanco(8000,827,'C',40e6,detecciones*Long);
z=distancias;

%%
delta=0.005;
xg = min(x)-delta : delta :max(x)+delta;
yg = min(y)-delta : delta :max(y)+delta;
zg = min(z)-delta : delta :max(z)+delta;
cubo=zeros(size(xg,2),size(yg,2),size(zg,2));
colores=zeros(size(xg,2),size(yg,2),size(zg,2));
minx=min(x);
miny=min(y);
minz=min(z);
minp=min(potencias);
for i = 1:size(x,2)
    cc=i;
    posx=floor((x(i)-minx)/delta)+1;
    posy=floor((y(i)-miny)/delta)+1;
    posz=floor((z(i)-minz)/delta)+1;
    cubo(posz,posx,posy)=1;
    colores(posz,posx,posy)=potencias(i);
end
figure
axis vis3d;
colormap copper
[faces,verts,colors] = isosurface(cubo,0.1,colores*100);
hpatch=patch('Vertices', verts, 'Faces', faces, ...
    'FaceVertexCData', colors, ...
    'FaceColor','interp', ...
    'edgecolor', 'interp');
```

Firmado: Alejandro Badolato Martín.

Ingeniero de Telecomunicación.

Madrid, 25 de Septiembre de 2013

