



UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES
DEPARTAMENTO DE ELECTRÓNICA

TÉCNICAS DIGITALES II

PROYECTO:

Analizador de Espectro de Audio

Año 2018

Grupo N° 5:

Integrantes:	De Rosa, Mauricio	140.508-1
	Hojnadel, Andrés	154.563-2
	Salazar Ato, Benjamín Yahir	155.803-1
	Soglio, Julián	141.448-3

Docentes: Ing. Alarcón, Juan Esteban
Ing. Crivelli, Marcelo Alberto



Índice

1. Descripción del proyecto	3
2. Diagrama en bloques	3
3. Especificaciones técnicas	4
4. Diagramas esquemáticos	4
4.1. Estructura jerárquica	4
4.2. Fuente de alimentación	5
4.3. Entrada de señal	6
4.4. Control Automático de Ganancia	7
4.5. ADC	8
4.6. μC	9
5. Lista de materiales (BOM)	10
5.1. Parte 1	10
5.2. Parte 2	11
6. PCB: Render 3D	12
7. Placa soldada	12
8. Pantalla: ILI9341	13
9. Software	13
9.1. Diagrama de flujo del programa	14
9.2. Código	14
9.2.1. Programa Principal: int main(void)	14
9.2.2. Inicialización del SO: void MX_FREERTOS_Init(void)	15
9.2.3. Tarea de lectura del teclado: void vTask_TecladoRead(void const *)	15
9.2.4. Tarea de resolución del teclado: void vTask_TecladoResolve(void const *)	16
9.2.5. Tarea de procesamiento: void vTask_SigProc(void const *)	16
10. Gabinete	17
10.1. Render 3D	17
10.2. Planos	18
10.2.1. Gabinete: dimensiones generales	18
10.2.2. Gabinete: detalle de vista frontal	19
10.2.3. Gabinete: perspectiva 1	20
10.2.4. Gabinete: perspectiva 2	20
10.2.5. Pieza superior: dimensiones generales	21
10.2.6. Pieza superior: perspectiva	22
11. Referencias	22

1. Descripción del proyecto

El proyecto a diseñar consta de un analizador de espectro de audio el cual podrá adquirir señales ya sea a través su micrófono interno incorporado o cualquier otro dispositivo externo mediante la entrada de audio de 3.5mm para luego realizarles una transformada rápida de Fourier (FFT)[1] y mostrar por pantalla el espectro de la misma en forma continua o en barras de octava. Además cuenta con un conector extra como salida de audio para poder reproducir el audio en el caso de utilizar un dispositivo externo. El ancho de banda a analizar es será de $1Hz$ a $20kHz$.

Internamente consta de cinco bloques: la fuente alimentación, la cual recibe a la entrada una tensión entre $9V$ y $12V$ y luego genera $5V$ y $3V3$ mediante dos fuentes *switching*; la etapa de adquisición y acondicionamiento de señal, formada por el conector de 3.5mm estereo y el micrófono para la adquisición, y un controlador automático de ganancia (AGC) para el acondicionamiento; un conversor analógico-digital (ADC PCM1802)[2] de 24 bits para realizar el muestreo; el microcontrolador (LPC1769)[3] encargado del procesamiento de la señal y comunicación de los periféricos; el display táctil por el cuál se mostrará la el espectro de la señal y que además funcionará como dispositivo de entrada para las configuraciones menores del dispositivo.

Como la alimentación es con fuente simple (no maneja tensiones negativas) en el acondicionamiento de señal se le agrega una tensión de offset de $2,5V$ tal que pueda excursionar de manera simétrica entre $0V$ y $5V$. Para ello, luego del levantamiento en continua, se utilizan amplificadores *rail-to-rail* para poder generar tensiones de salida de tensión similar a su alimentación.

2. Diagrama en bloques

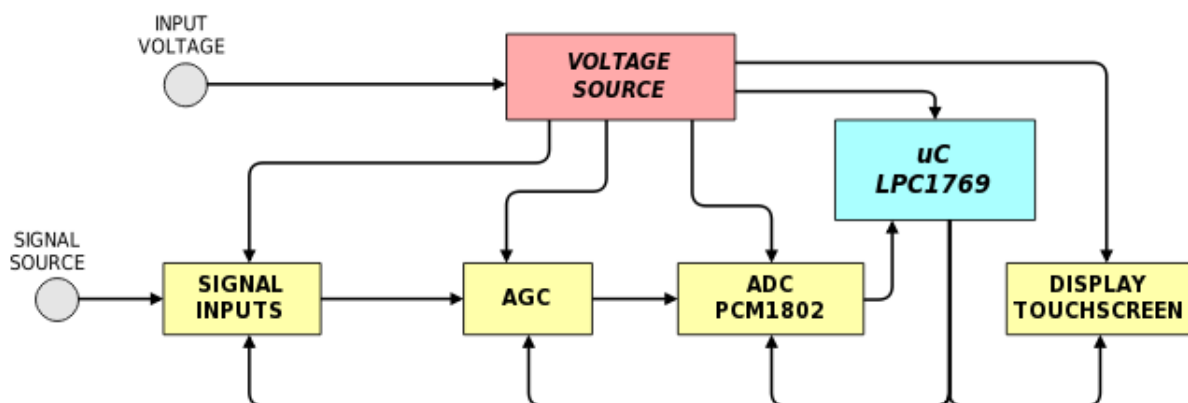


Figura 1: Diagrama en bloques

3. Especificaciones técnicas

- Tensión de alimentación: $9V \sim 12V$
- Amplitud máxima de señal: $1,5V_{pp}$
- Ancho de banda: $20kHz$
- Display: TFT LCD Touchscreen 3.2"

4. Diagramas esquemáticos

Todos los diagramas esquemáticos del dispositivo y el diseño del PCB fue realizado mediante el software *Altium*[4].

4.1. Estructura jerárquica

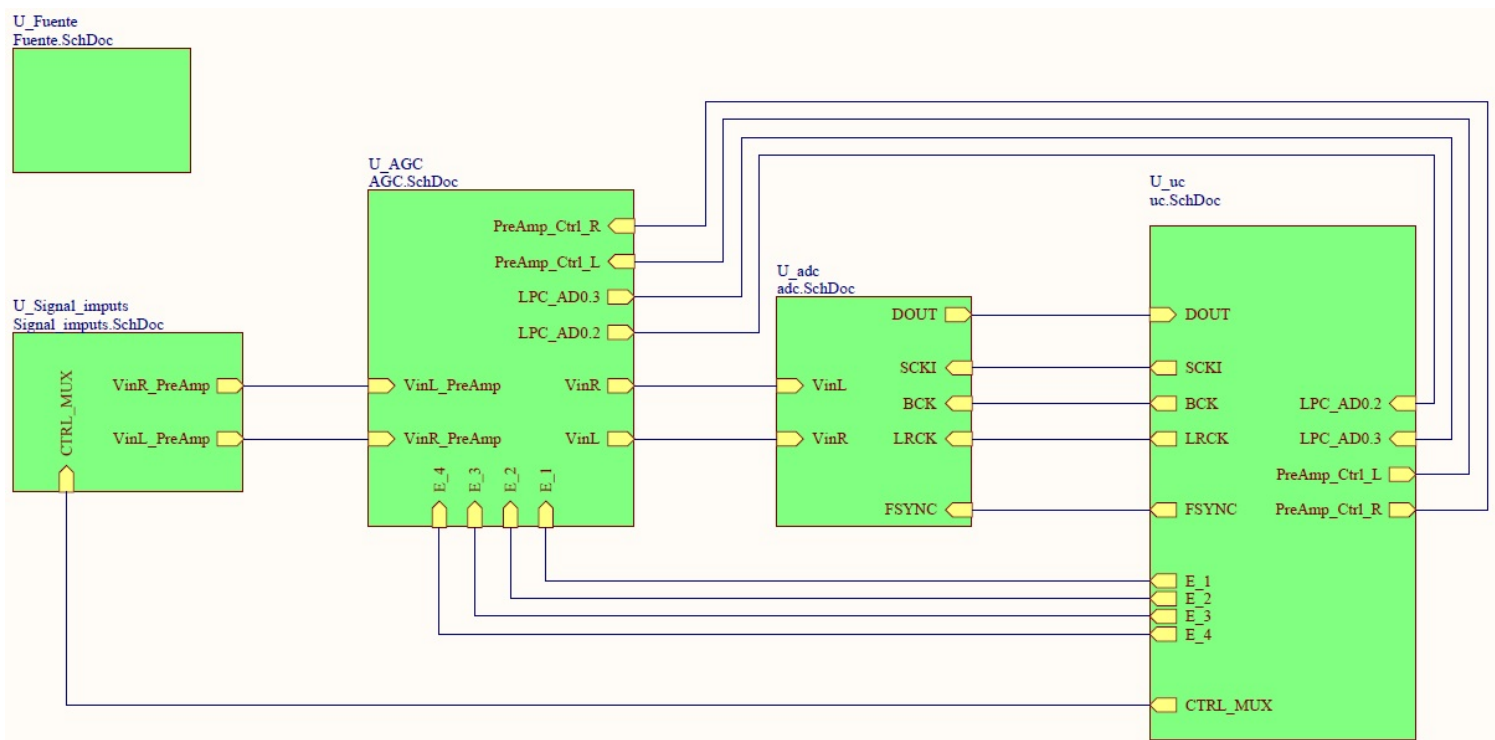


Figura 2: Estructura jerárquica

4.2. Fuente de alimentación

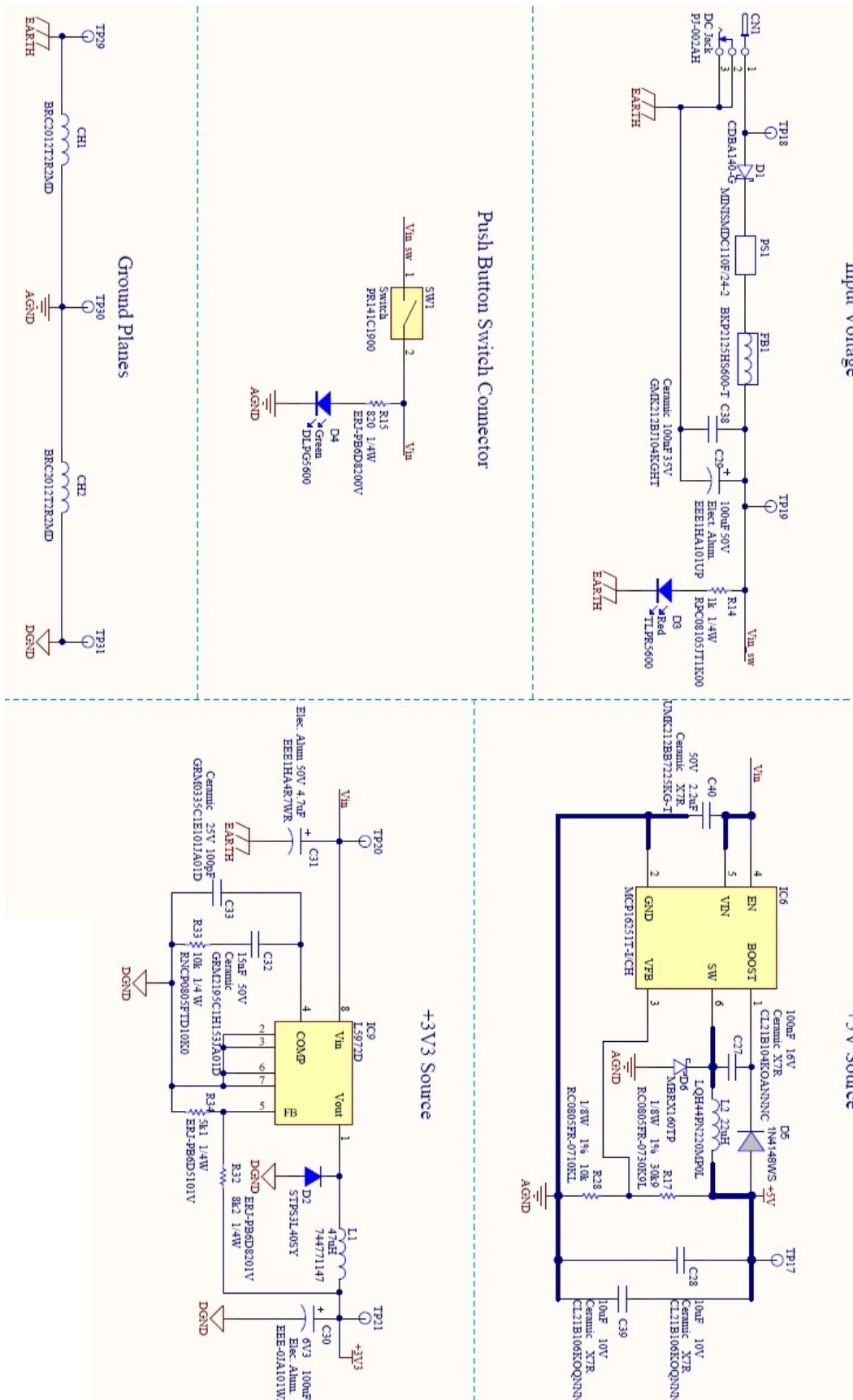


Figura 3: Esquemático de la fuente de alimentación

4.3. Entrada de señal

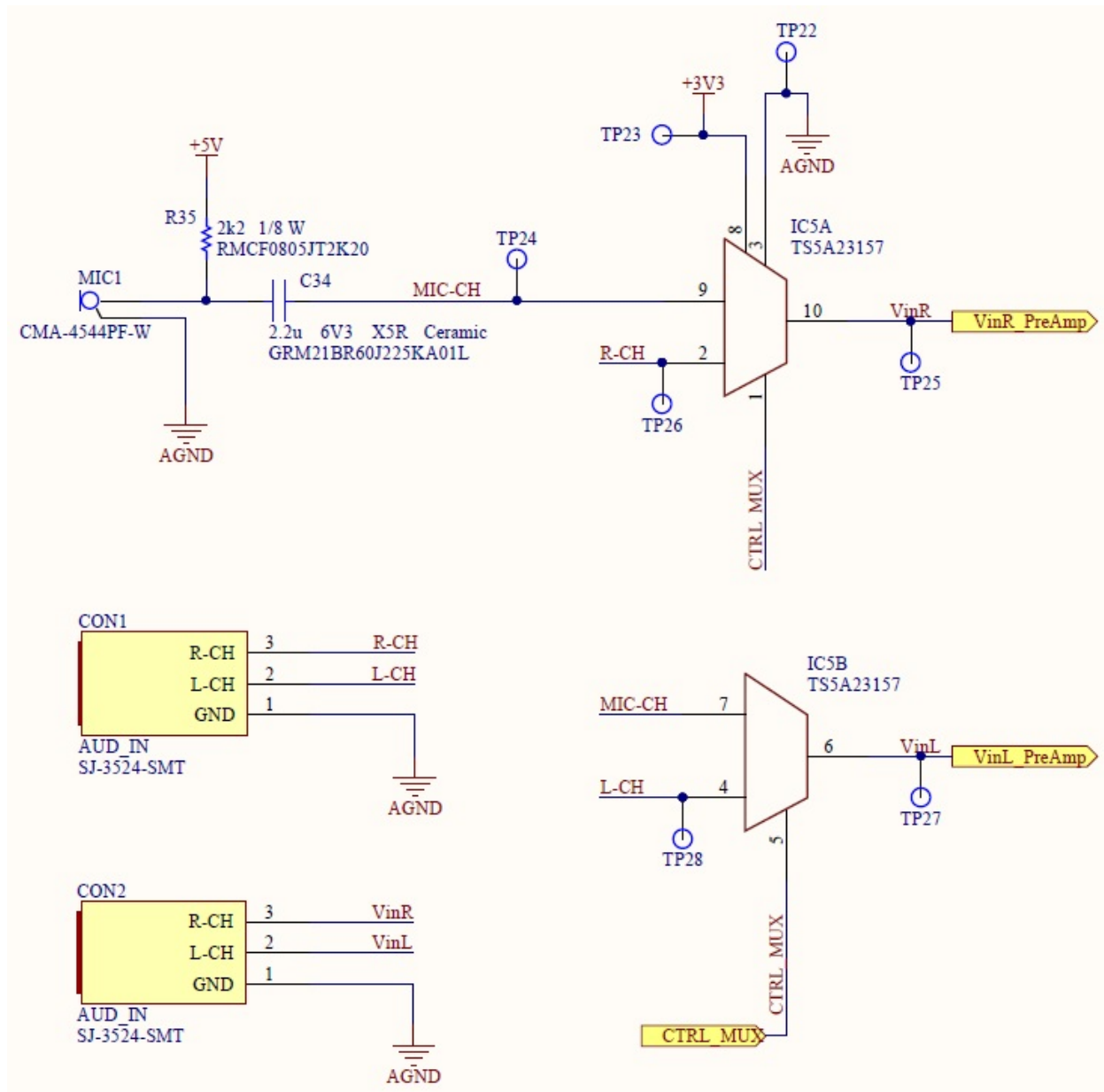


Figura 4: Esquemático de las entradas de señal



4.5. ADC

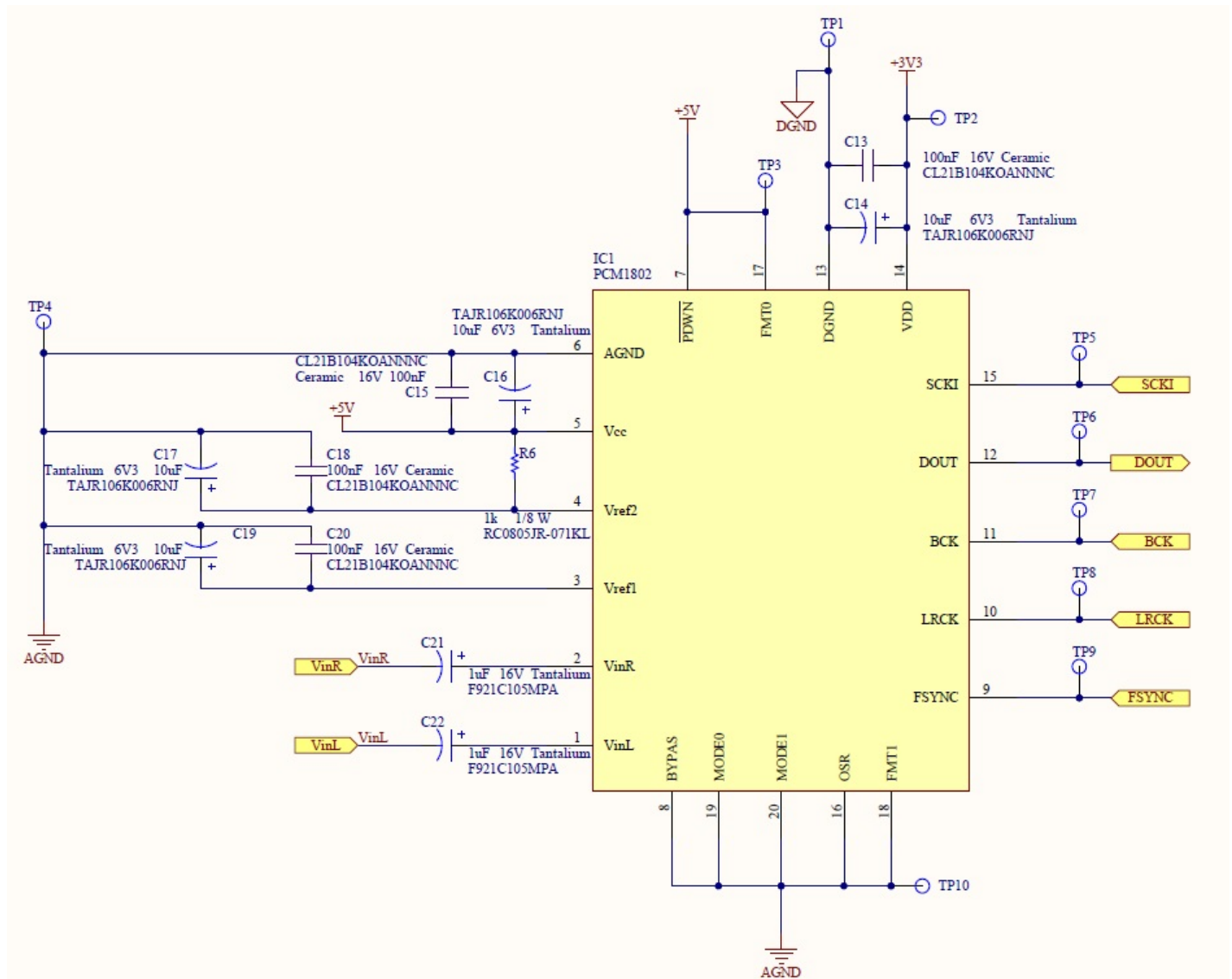
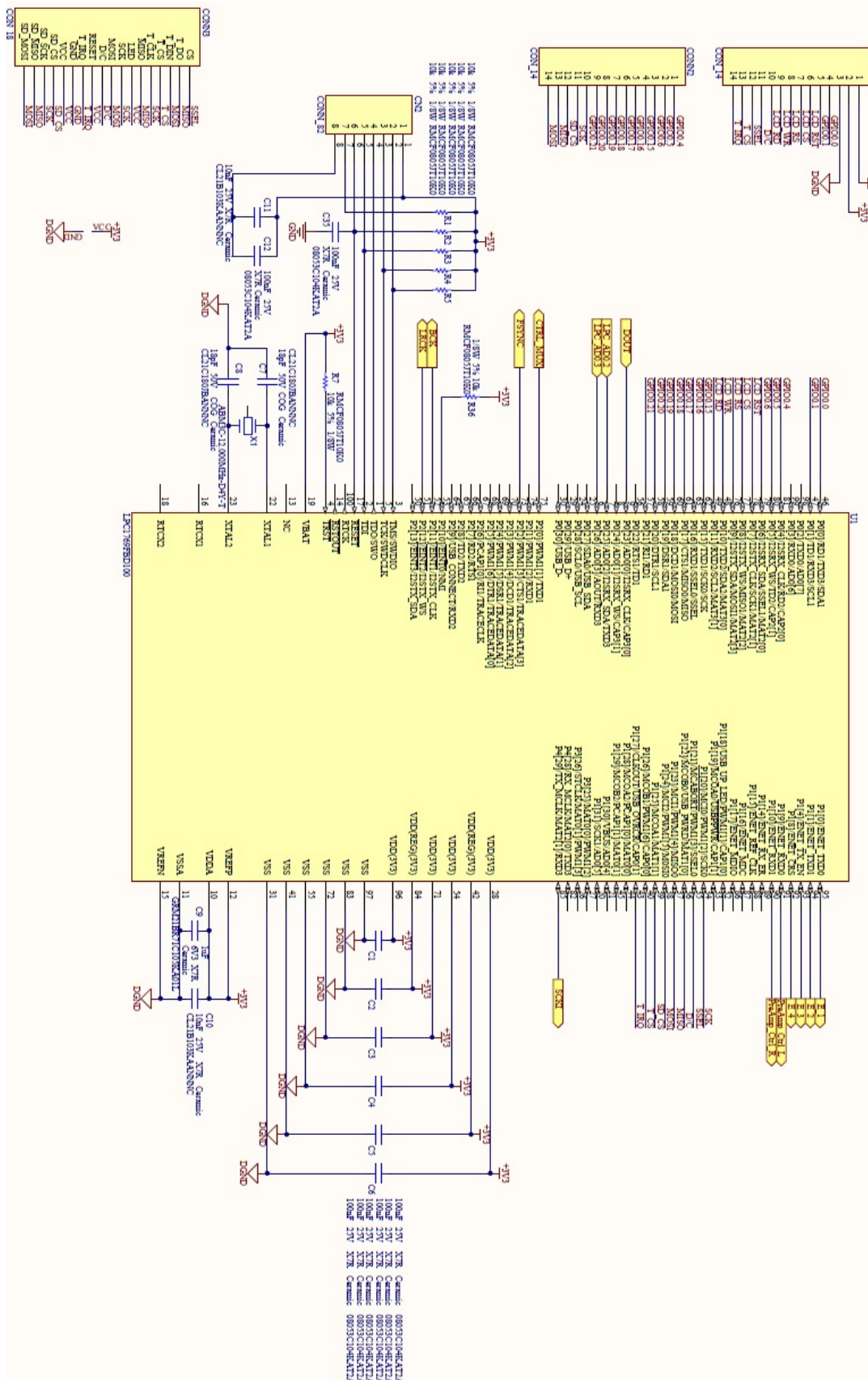


Figura 6: Esquemático del ADC

4.6. μC





5. Lista de materiales (BOM)

5.1. Parte 1

#	Description	Comment	Designator	Part Number	Quantity	C	Temp Coef	V	Value	Price un. (USD)	Total (USD)
1	Analog Switch	TS5A23157	IC4, IC5	TS5A23157DGSR	2	-	-	-	-	0,92	1,84
2	Connector	DC Jack	CN1	PJ-002AH	1	-	-	-	-	0,74	0,74
3	Connector	CONN 82	CN2	-Tira de pines-	1	-	-	-	-	-	-
4	CONN TFT 3.5	CONN 14	CONN1, CONN2	-Tira de pines-	2	-	-	-	-	-	-
5	CONN TFT 2.8	CONN 18	CONN3	-Tira de pines-	1	-	-	-	-	-	-
6	Capacitor	100nF 25V X7R Ceramic	C1, C2, C3, C4, C5, C6, C12, C35	08053C104KAT2A	8	100nF	X7R	25V	-	0,10	0,80
7	Capacitor	18pF 50V COG Ceramic	C7, C8	CL21C180JBANNNC	2	18pF	COG	50V	-	0,10	0,20
8	Capacitor	1uF 6V3 X7R Ceramic	C9	GRM21BR71C105KA01L	1	1uF	X7R	6V3	-	0,11	0,11
9	Capacitor	10nF 25V X7R Ceramic	C10, C11	CL21B103KAANNNC	2	10nF	X7R	25V	-	0,10	0,20
10	Capacitor	100nF 16V Ceramic	C13, C15, C18, C20	CL21B104KOANNNC	4	100nF	-	16V	-	0,10	0,40
11	Capacitor	1uF 16V Ceramic	C23, C26	GRM21BR71C105KA01L	2	1uF	-	16V	-	0,11	0,22
12	Capacitor	100nF 16V X7R	C27	CL21B104KOANNNC	1	100nF	-	16V	-	0,10	0,10
13	Capacitor	10uF 10V X7R	C28, C39	CL21B106KOANNNG	2	10uF	-	10V	-	0,29	0,58
14	Capacitor	15nF 50V Ceramic	C32	GRM2195C1H153JA01D	1	15nF	-	50V	-	0,33	0,33
15	Capacitor	100pF 25V Ceramic	C33	C0805C101K3GACTU	1	100pF	-	25V	-	0,20	0,20
16	Capacitor	2.2u 6V3 X5R Ceramic	C34	GRM21BR60J225KA01L	1	2.2u	X5R	6V3	-	0,20	0,20
17	Capacitor	100nF 35V Ceramic	C38	GMMK212BJ104KGT	1	100nF	-	35V	-	0,14	0,14
18	Capacitor	2.2uF 50V X7R	C40	UMK212BB7225KG-T	1	2.2uF	X7R	50V	-	0,28	0,28
19	Capacitor	10uF 6V3 Tantalum	C14, C16, C17, C19	TAJUR106K006RNU	4	10uF	-	6V3	-	1,02	4,08
20	Capacitor	1uF 16V Tantalum	C21, C22	F921C105MPA	2	1uF	-	16V	-	0,35	0,70
21	Capacitor	100uF 16V Elect. Alum.	C24, C25, C36, C37	UWX1C101MCL1GB	4	100uF	-	16V	-	0,33	1,32
22	Capacitor	100uF 50V Elecaalum	C29	EEE1HA101UP	1	100uF	-	50V	-	0,59	0,59
23	Capacitor	100uF 6V3 Elecaalum	C30	EEE-0JA101WR	1	100uF	-	6V3	-	0,31	0,31
24	Capacitor	4.7uF 50V Elecaalum	C31	EEE1HA4R7WR	1	4.7uF	-	50V	-	0,34	0,34
25	Choke	Choke	CH1, CH2	BRC201212R2MD	2	-	-	-	-	0,34	0,68
26	Diode	STPS3L40S	D2	STPS3L40SY	1	-	-	-	-	0,93	0,93
27	Diode signal	1N4148WS	D5	1N4148WS	1	-	-	-	-	0,17	0,17
28	Diode Schottky	CDBA140-G	D1	CDBA140-G	1	-	-	-	-	0,37	0,37
29	Diode Rectificador	MBRX160TP	D6	MBRX160TP	1	-	-	-	-	0,45	0,45
30	Ferrite Bead	Ferrite Bead	FB1	BKP2125HS600-T	1	-	-	-	-	0,10	0,10



5.2. Parte 2

#	Description	Comment	Designator	Part Number	Quantity	C	Temp Coef	V	Value	Price un. (USD)	Total (USD)
31	Inductor	47uH	L1	744771147	1	-	-	-	47uH	2.12	2.12
32	Inductor	22uF	L2	LQH44PN220MP0L	1	-	-	-	22uH	0.43	0.43
33	Connector	AUD_IN	CON1, CON2	SJ-3524-SMT	2	-	-	-	-	1.04	2.08
34	Fuente Switching	L5972D	IC9	L5972D013TR	1	-	-	-	-	2.28	2.28
35	LED	LED	D3	TLPR5600	1	-	-	-	-	0.53	0.53
36	CPU Cortex M3	LPC1769FBD100	U1	LPC1769FBD100	1	-	-	-	-	11.53	11.53
37	Microfono	Mic	MIC1	CMA-4544PF-W	1	-	-	-	-	0.82	0.82
38	MCP6022	MCP6022	IC2	MCP6022T-I/SNCT-ND	1	-	-	-	-	1.38	1.38
39	PCM1802	PCM1802	IC1	PCM1802DBR	1	-	-	-	-	3.88	3.88
40	Poliswitch	Polyswitch	PS1	MINISMD C110F/24.2	1	-	-	-	-	0.63	0.63
41	74LVC4066	74LVC4066	IC3, IC7	74LVC4066BQ,115	2	-	-	-	-	0.63	1.26
42	Resistor	10k 5% 1/8W	R1, R2, R3, R4, R5, R7, R36	RMCF0805JT10K0	7	-	-	-	10k	0.10	0.70
43	Resistor	1k 1/8W	R6, R19, R31	RC0805JR-071KL	3	-	-	-	1k	0.10	0.30
44	Resistor	100k 1/8W	R8, R9, R10, R18, R20, R21, R23, R27	RC0805FR-07100KL	8	-	-	-	100k	0.10	0.80
45	Resistor	590 1/8W	R11, R22	RC0805FR-07590RL	2	-	-	-	590	0.10	0.20
46	Resistor	33k 1/8W	R12, R26	RMCF0805JT33K0	2	-	-	-	33k	0.10	0.20
47	Resistor	2k2 1/8W	R13, R24, R35	RMCF0805JT2K20	3	-	-	-	2k2	0.10	0.30
48	Resistor	1k 1/4W	R14	LTR10EZPJ102	1	-	-	-	1k	0.17	0.17
49	Resistor	820 1/4W	R15	ERJ-PB6D8200V	1	-	-	-	820	0.24	0.24
50	Resistor	3k9 1/8W	R16, R25	RMCF0805JT3K90	2	-	-	-	3k9	0.10	0.20
51	Resistor	30k9 1/8W 1%	R17	RC0805FR-0730K9L	1	-	-	-	30k9	0.10	0.10
52	Resistor	8k2 1/4W	R32	ERJ-PB6D8201V	1	-	-	-	8k2	0.10	0.10
53	Resistor	10k 1/4W	R33	RNCP0805FTD10K0	1	-	-	-	10k	0.10	0.10
54	Resistor	5k1 1/4W	R34	ERJ-PB6D5101V	1	-	-	-	5k1	0.24	0.24
55	Switch	Switch	SW1	PR141C1900	1	-	-	-	-	1.07	1.07
56	Switching Source	MCP	IC6	MCP1625T1-I/CH	1	-	-	-	-	0.60	0.60
57	Crystal	XTAL_HF	X1	ABM3C-12.000MHz-D4Y-T	1	-	-	-	-	0.69	0.69
58	LED	LED	D4	DLPG5600	1	-	-	-	-	0.57	0.57
59	Resistor	10k 1/8W 1%	R28	RC0805FR-0710KL	1	-	-	-	10k	0.10	0.10
60											
Total (USD)											50.00

6. PCB: Render 3D

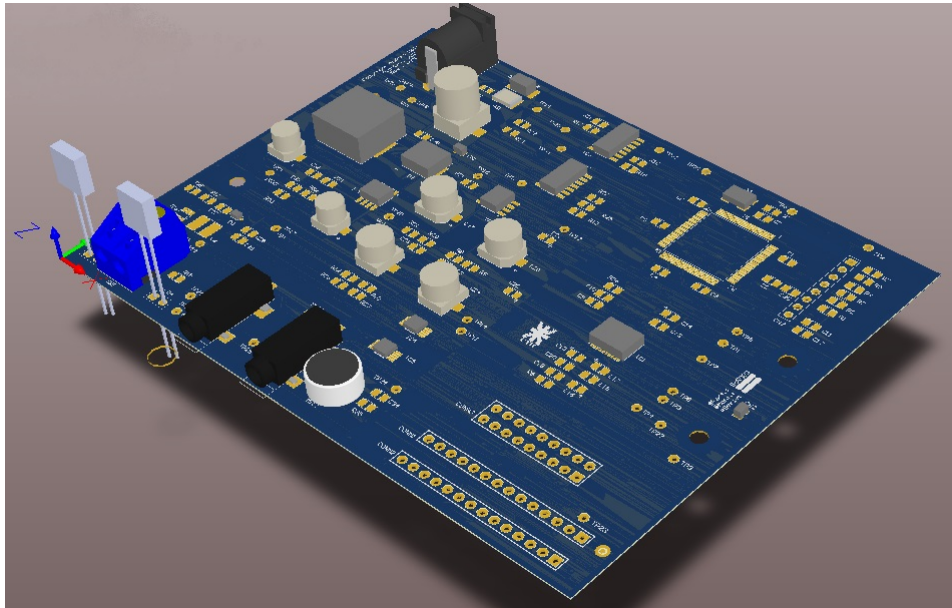


Figura 8: Render 3D del PCB

7. Placa soldada

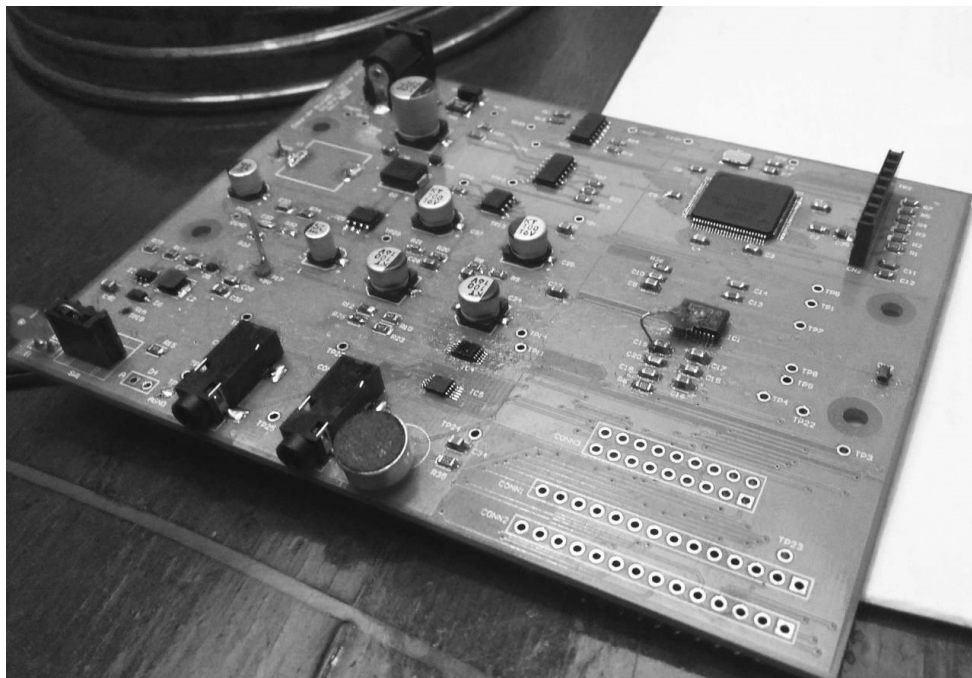
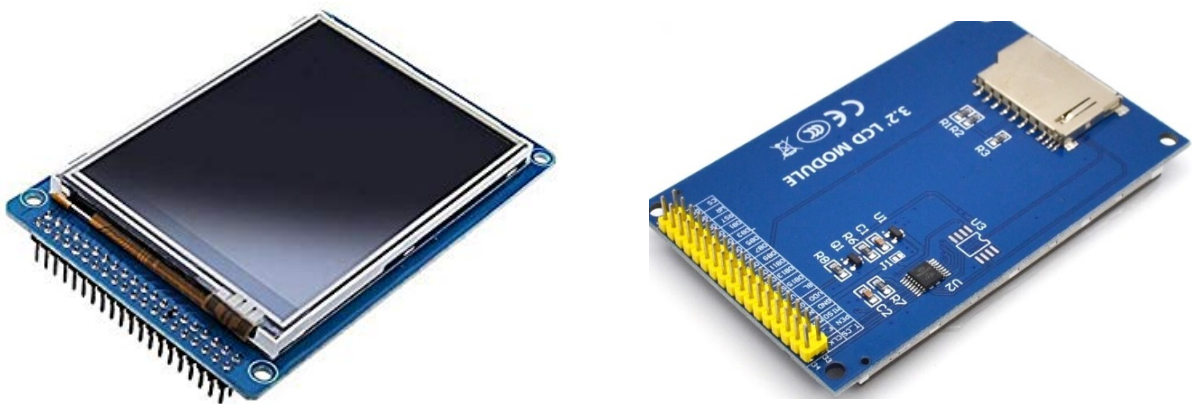


Figura 9: Fotografía de la placa con componentes soldados

8. Pantalla: ILI9341

La pantalla utilizada es una pantalla TFT (Thin Film Transistor) de 240x320 píxeles RGB, modelo ILI9341[5]. Utiliza alimentación de 5V para manejar la iluminación de background y 3.3V para las señales de comunicación. Cuenta con 34 pines distribuidos en dos filas de 17 que se conectan al PCB mediante un bus plano.

La escritura en la pantalla puede realizarse mediante comunicación paralelo o SPI. Posee interacción táctil y un slot para tarjeta de memoria SD las cuales también se controlan mediante protocolo SPI.



9. Software

El software desarrollado para este dispositivo se ejecuta en el microcontrolador LPC-1769 de la familia ARM Cortex-M [8] (Cortex-M3). Del dispositivo se utilizan los periféricos de ADC (2 canales) para implementar un AGC (control automático de ganancia), el periférico de I2S como protocolo de comunicación entre el μC y el ADC externo, DMA para almacenar los datos convertidos por este ADC, el periférico de SPI para la comunicación del *touchscreen* y de la memoria SD.

El código fue desarrollado en lenguaje C mediante el entorno de desarrollo que proporciona la firma del μC : *LPC-Expresso*[6].

La lógica del programa es administrada por el sistema operativo en tiempo real para sistemas embebidos *FreeRTOS*[7].

Para el desarrollo fueron necesarias la biblioteca *arm_math* de la CMSIS-DSP[9] para las operaciones relacionadas al procesamiento de la señal, y la biblioteca *TFT_ILI9341* para el manejo de la pantalla.

9.1. Diagrama de flujo del programa

El dispositivo recibe una señal analógica estéreo la cual es convertida a formato digital mediante el ADC PCM-1802 a una frecuencia de muestreo de 32kHz y 32 bits por muestra. Los bits de la señal son mandados mediante protocolo I2C en formato Q1.31 y son almacenados en memoria utilizando DMA. Se disponen de un buffer de memoria dividido en dos mitades de manera de que cuando se completa la primer mitad, ésta se procesa y se siguen almacenando datos en la segunda mitad para luego repetir el proceso.

El procesamiento consta de la suma de ambos canales con su previo escalado, se aplica la ventana de muestreo definida por el usuario (rectangular, Hamming o Blackman) y luego se realiza el algoritmo de FFT. Una vez obtenido el módulo de la FFT, se la grafica en pantalla según el span de frecuencia definido también por el usuario.

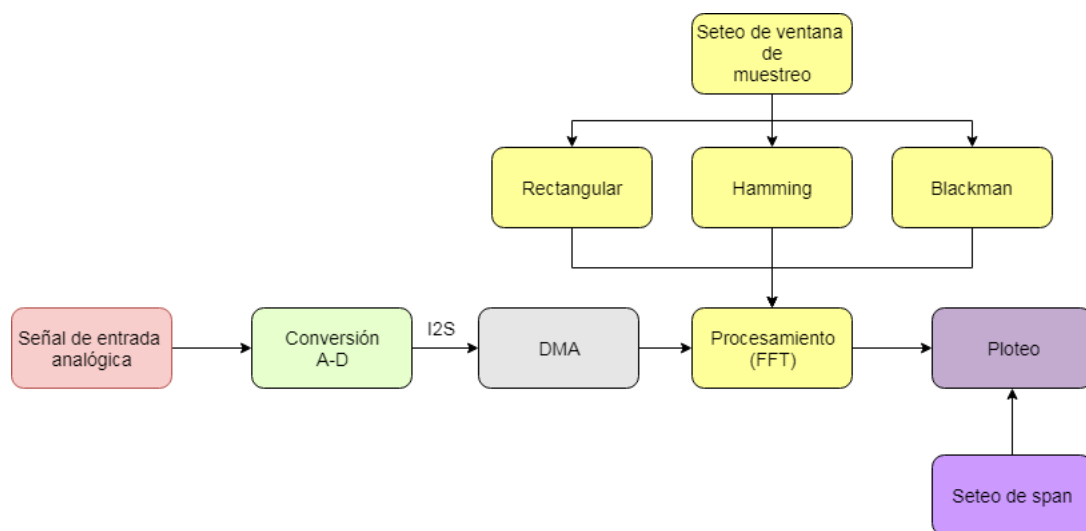


Figura 10: Diagrama de flujo del programa

9.2. Código

9.2.1. Programa Principal: int main(void)

```
1 int main(void){
2     HAL_Init();
3     SystemClock_Config();
4
5     MX_GPIO_Init();
6     MX_DMA_Init();
7     MX_I2S2_Init();
8     MX_I2S3_Init();
9     MX_SPI1_Init();
10    MX_ADC2_Init();
11    MX_ADC1_Init();
12    init_main();
13
14    arm_rfft_fast_init_f32(&sint,MUESTRAS);
15    TM_ILI9341_MPX_axis();
16
17    MX_FREERTOS_Init();
18
19    osKernelStart();
20
21    while (1){
22    }
23 }
```



9.2.2. Inicialización del SO: void MX_FREERTOS_Init(void)

```
1 void MX_FREERTOS_Init(void) {
2
3
4     /* Create the semaphores(s) */
5     /* definition and creation of SemRead */
6     osSemaphoreDef(SemRead);
7     SemReadHandle = osSemaphoreCreate(osSemaphore(SemRead), 1);
8
9     /* definition and creation of SemProc */
10    osSemaphoreDef(SemProc);
11    SemProcHandle = osSemaphoreCreate(osSemaphore(SemProc), 1);
12
13    /* definition and creation of SemPlot */
14    osSemaphoreDef(SemPlot);
15    SemPlotHandle = osSemaphoreCreate(osSemaphore(SemPlot), 1);
16
17    /* Create the thread(s) */
18    /* definition and creation of defaultTask */
19    osThreadDef(defaultTask, StartDefaultTask, osPriorityNormal, 0, 128);
20    defaultTaskHandle = osThreadCreate(osThread(defaultTask), NULL);
21
22    /* definition and creation of I2SRead */
23    osThreadDef(I2SRead, vTask_I2SRead, osPriorityNormal, 0, 128);
24    I2SReadHandle = osThreadCreate(osThread(I2SRead), NULL);
25
26    /* definition and creation of SigProc */
27    osThreadDef(SigProc, vTask_SigProc, osPriorityBelowNormal, 0, 128);
28    SigProcHandle = osThreadCreate(osThread(SigProc), NULL);
29
30    /* definition and creation of Plot */
31    osThreadDef(Plot, vTask_Plot, osPriorityLow, 0, 128);
32    PlotHandle = osThreadCreate(osThread(Plot), NULL);
33
34    /* definition and creation of TecladoRead */
35    osThreadDef(TecladoRead, vTask_TecladoRead, osPriorityNormal, 0, 128);
36    TecladoReadHandle = osThreadCreate(osThread(TecladoRead), NULL);
37
38    /* definition and creation of TecladoResolve */
39    osThreadDef(TecladoResolve, vTask_TecladoResolve, osPriorityNormal, 0, 128);
40    TecladoResolveHandle = osThreadCreate(osThread(TecladoResolve), NULL);
41
42    /* Create the queue(s) */
43    /* definition and creation of QueueKey */
44    osMessageQDef(QueueKey, 1, uint8_t);
45    QueueKeyHandle = osMessageCreate(osMessageQ(QueueKey), 0);
46
47    /* definition and creation of QueueWindow */
48    osMessageQDef(QueueWindow, 1, uint8_t);
49    QueueWindowHandle = osMessageCreate(osMessageQ(QueueWindow), 0);
50
51    /* definition and creation of QueueSpan */
52    osMessageQDef(QueueSpan, 1, uint8_t);
53    QueueSpanHandle = osMessageCreate(osMessageQ(QueueSpan), 0);
54 }
```

9.2.3. Tarea de lectura del teclado: void vTask_TecladoRead(void const *)

```
1 void vTask_TecladoRead(void const *){
2
3     if ( !HAL_GPIO_ReadPin(SW2_GPIO_Port, SW2_Pin) )
4         xQueueOverwrite(QueueKey, SW1);
5     else if ( !HAL_GPIO_ReadPin(SW1_GPIO_Port, SW1_Pin) )
6         xQueueOverwrite(QueueKey, SW2);
7     else if ( !HAL_GPIO_ReadPin(SW3_GPIO_Port, SW3_Pin) )
8         xQueueOverwrite(QueueKey, SW3);
9     else if ( !HAL_GPIO_ReadPin(SW4_GPIO_Port, SW4_Pin) )
10        xQueueOverwrite(QueueKey, SW4);
11
12 }
```




9.2.4. Tarea de resolución del teclado: void vTask_TecladoResolve(void const *)

```
1 void vTask_TecladoResolve(void const *){
2
3     uint8_t tecla = NOKEY;
4     uint8_t tipo_ventana = RECTANGULAR;
5     uint8_t span = FULL.SPAN;
6
7     while(1){
8
9         xQueueReceive(QueueKey, &tecla, portMAX_TICK);
10
11         switch (tecla){
12             case SW1:
13                 span = span + SCREEN.WIDTH - FFT.OFFSET.X;
14                 if (span == RANGO_871_1160 + SCREEN.WIDTH - FFT.OFFSET.X)
15                     span = 0;
16                 TM_ILI9341_MPX_axis_setup(span);
17                 xQueueOverwrite(QueueSpan, &span);
18                 break;
19
20             case SW2:
21                 tipo_ventana++;
22                 if (tipo_ventana > 2)
23                     tipo_ventana = 0;
24                 xQueueOverwrite(QueueWidnow, &tipo_ventana);
25                 Nombre_ventana_pantalla (tipo_ventana);
26                 break;
27
28             case SW3:
29                 break;
30
31             case SW4:
32                 break;
33
34             //End Switch
35
36         tecla = NO_TECLA;
37     }
38 }
```

9.2.5. Tarea de procesamiento: void vTask_SigProc(void const *)

```
1 void vTask_SigProc(void const * argument)
2 {
3     uint8_t window_type = RECTANGULAR;
4     uint32_t index;
5     float32_t max=0;
6
7     xSemaphoreTake(SemProcHandle, portMAX_DELAY);
8
9     for (;;) {
10         xSemaphoreTake(SemProcHandle, portMAX_DELAY);
11
12         arm_scale_f32(buffer_rx_left, FFT_SCALE, buffer_rx_left, LEN_TX_RX);
13         arm_scale_f32(buffer_rx_right, FFT_SCALE, buffer_rx_right, LEN_TX_RX);
14
15         arm_add_f32(buffer_rx_left, buffer_rx_right, buffer_suma, LEN_TX_RX); // suma = buffer 1
16
17         xQueuePeek(QueueWindow, &window_type, NO_WAIT);
18
19         switch (window_type){
20             case (RECTANGULAR):
21                 break;
22             case (BLACKMAN):
23                 arm_mult_f32(buffer_suma, blackman_window, buffer_suma, MUESTRAS);
24                 break;
25             case (HAMMING):
26                 arm_mult_f32(buffer_suma, hamming_window, buffer_suma, MUESTRAS);
27                 break;
28         }
29     }
```

```
29  
30     arm_rfft_fast_f32(&sint , buffer_suma , transformada , 0);  
31  
32     arm_max_f32(transformada , MUESTRAS, &max, &index);  
33  
34     if (max < FFT.SATURATION){  
35  
36         arm_copy_f32(transformada_final , transformada_final_anterior , MUESTRAS/2);  
37  
38         arm_cmplx_mag_f32(transformada , transformada_final , MUESTRAS);  
39     }  
40  
41     xSemaphoreGive(SemPlotHandle);  
42  
43 }  
44 }
```

10. Gabinete

El diseño del gabinete fue realizado mediante la herramienta *FreeCAD*[10], la cual permite importar los modelos 3D del PCB diseñado en *Altium*, como así también de la pantalla a utilizar.

10.1. Render 3D

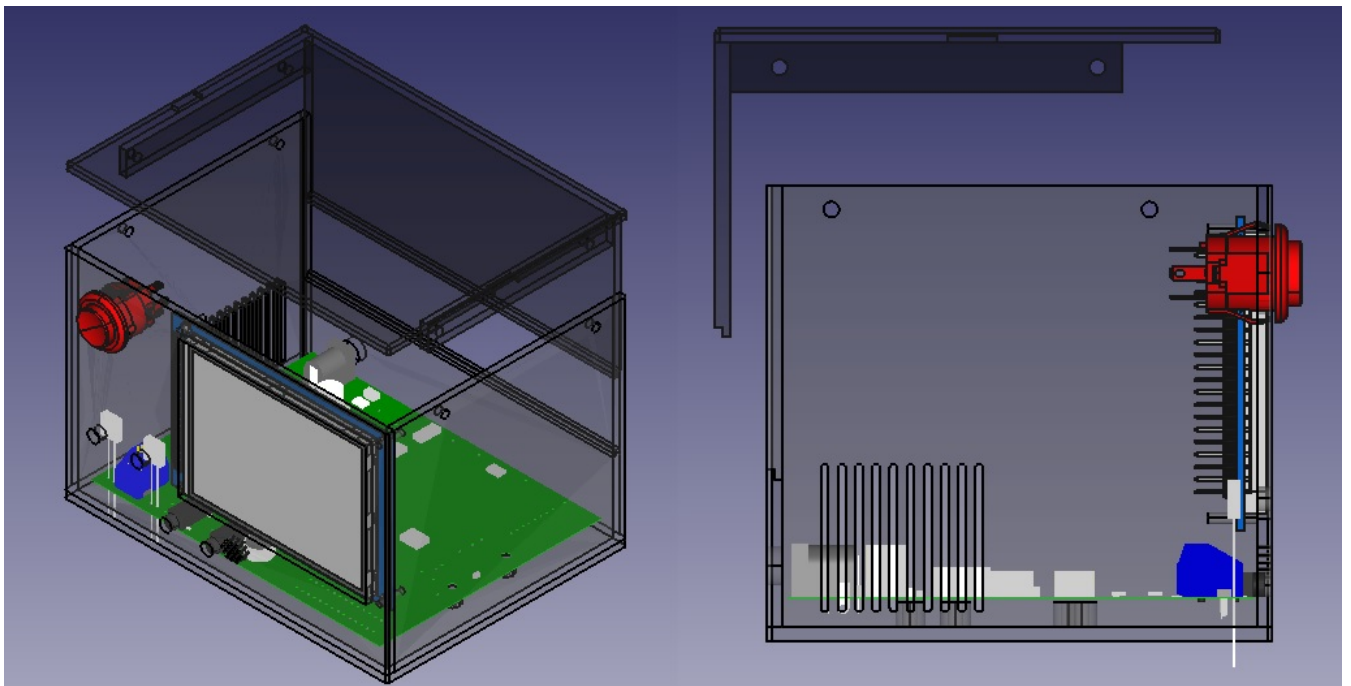
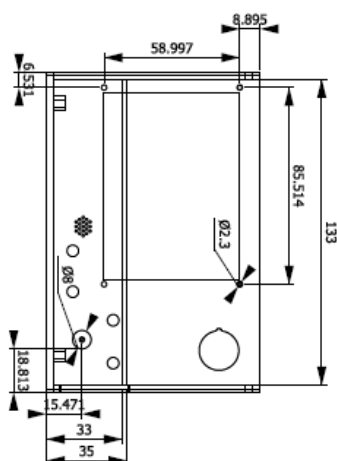
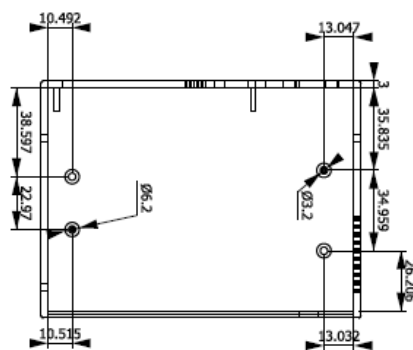
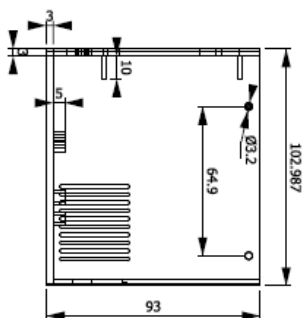
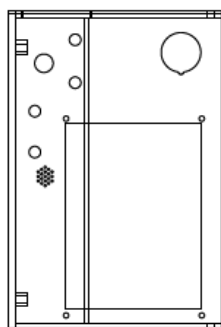
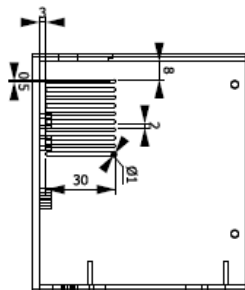


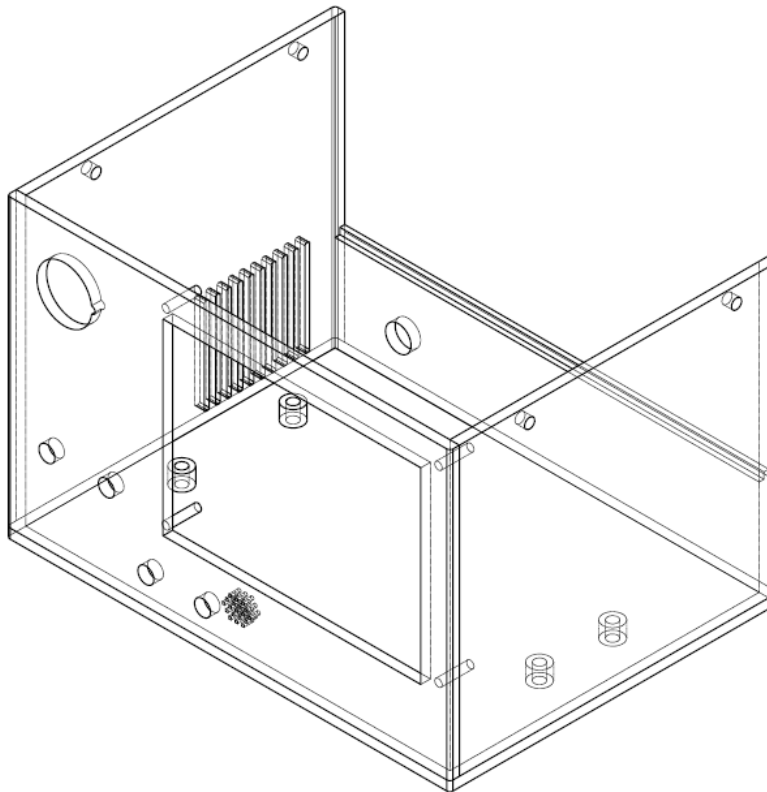
Figura 11: Vista 3D del gabinete

10.2. Planos

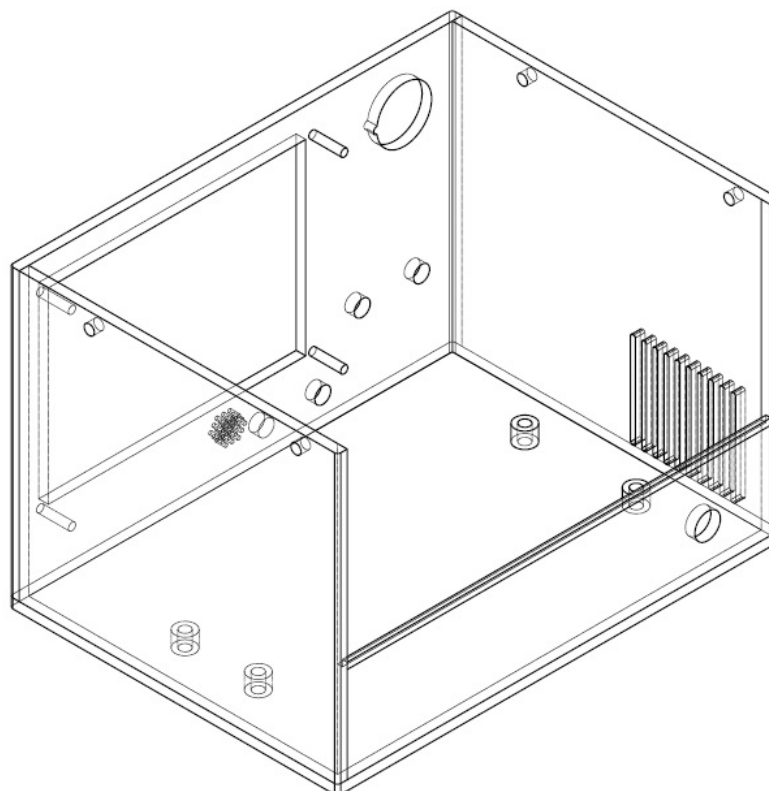
10.2.1. Gabinete: dimensiones generales



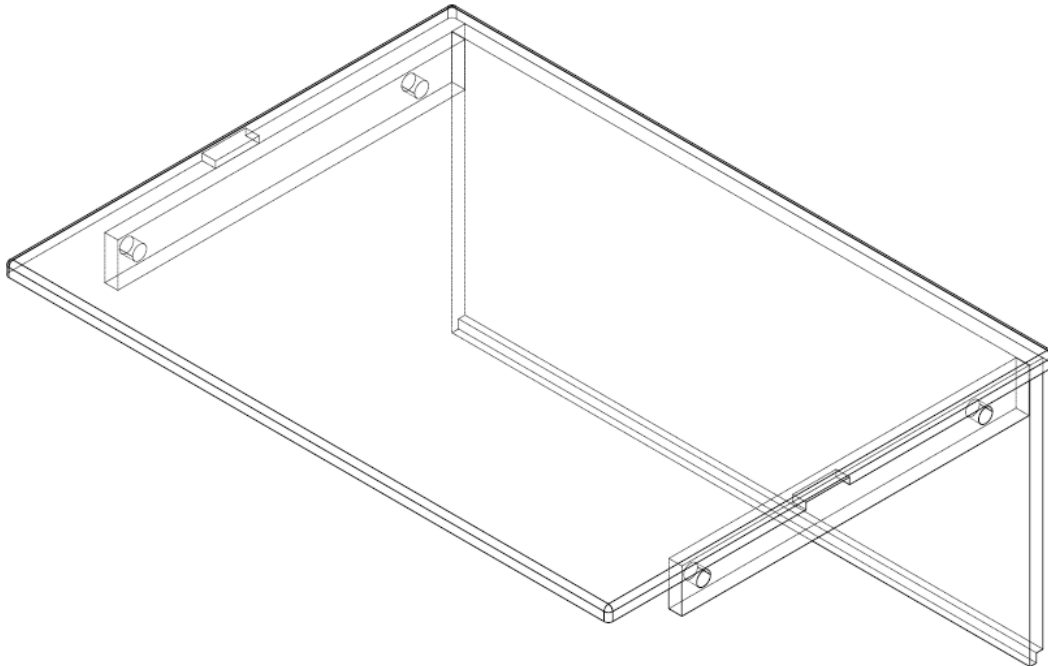
10.2.3. Gabinete: perspectiva 1



10.2.4. Gabinete: perspectiva 2



10.2.6. Pieza superior: perspectiva



Referencias

- [1] FFT (Fast Fourier Transform). https://en.wikipedia.org/wiki/Fast_Fourier_transform.
- [2] PCM 1802. <https://www.ti.com/lit/ds/symlink/pcm1802.pdf>
- [3] NXP LPC-1769. <https://www.nxp.com/docs/en/data-sheet/LPC1769-68-67-66-65-64-63.pdf>
- [4] Altium. <https://www.altium.com/es>
- [5] ILI9341 TFT Touchscreen 3.2". <https://cdn-shop.adafruit.com/datasheets/ILI9341.pdf>
- [6] LPCExpresso. <https://www.nxp.com/products/processors-and-microcontrollers/arm-based-processors-and-mcus/lpc-cortex-m-mcus/lpc1100-cortex-m0-plus-m0/lpcxpresso-ide-v8.2.2:LPCXPRESSO>
- [7] FreeRtos. <https://www.freertos.org/>
- [8] ARM Cortex-M. <https://www.arm.com/products/silicon-ip-cpu>
- [9] CMSIS-DSP. <http://www.keil.com/pack/doc/CMSIS/DSP/html/index.html>
- [10] FreeCAD. <https://www.freecadweb.org/>