



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

EN VÍAS DE DESARROLLO DEL SOFTWARE Y CALIBRACIÓN DEL BACKEND
DIGITAL DEL NUEVO PROTOTIPO DEL PAYLOAD “LANGMUIR PROBE” PARA
EL PROYECTO SUCHAI.

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO

MATÍAS CARLOS SANDOVAL NAVARRETE

PROFESOR GUÍA:
MARCOS ANDRES DÍAZ QUEZADA

CO-GUÍA:
MIGUEL MARTÍNEZ LEDESMA

MIEMBRO DE LA COMISIÓN:
ANDRÉS CABA RUTTE

SANTIAGO DE CHILE
2020

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO
POR: MATÍAS CARLOS SANDOVAL NAVARRETE
FECHA: 2020
PROF. GUÍA: MARCOS ANDRES DÍAZ QUEZADA

EN VÍAS DE DESARROLLO DEL SOFTWARE Y CALIBRACIÓN DEL BACKEND
DIGITAL DEL NUEVO PROTOTIPO DEL PAYLOAD “LANGMUIR PROBE” PARA
EL PROYECTO SUCHAI.

Uno de los experimentos principales del satélite SUCHAI-I es el “Langmuir Probe” o medidor de densidad de plasma, el cual mide la densidad de partículas cargadas presentes en la ionosfera a lo largo de su camino orbital. Este instrumento tiene cuatro partes principales: el captador de partículas (o electrodo), la etapa análoga, la etapa digital y la etapa de estimación de la densidad de partículas.

En el SUCHAI-1 el sensor de Langmuir capturaba electrones al utilizar un voltaje fijo positivo y tenía un electrodo esférico. Estas dos características dificultaron la estimación de la densidad de electrones. El uso de un voltaje positivo hace que la estimación de la densidad de electrones se degrade con el tiempo, debido al efecto de “potencial de carga” que se produce, debido a que el satélite se va cargando con electrones con el tiempo. Un electrodo esférico en principio ayuda a que el satélite no deba tener una estimación y control de orientación. Sin embargo, para una correcta medición con el electrodo esférico, este debía ser desplegado, idealmente varias decenas de centímetros fuera del satélite para evitar las perturbaciones que el chasis metálico produce, lo que impuso varias dificultades mecánicas.

Para evitar estas dificultades el diseño ideado para las siguientes misiones, los SUCHAI-2 y -3, utiliza un electrodo plano y un voltaje negativo. El electrodo plano no es necesario desplegarlo fuera del satélite, se puede ubicar en alguna cara del vehículo, facilitando la parte mecánica del instrumento. Sin embargo, esto implica que el satélite deba estimar y controlar su orientación, asegurando que el electrodo plano vaya en la parte delantera respecto a la dirección de viaje del satélite. Al usar un voltaje negativo el instrumento estima los iones en lugar de los electrones, lo que lo hace inmune al fenómeno de potencial de carga. Si el plasma se presume en equilibrio termodinámico, entonces la densidad de iones es igual a la densidad de electrones.

Este trabajo se concentra en el diseño, implementación y prueba de la etapa digital del nuevo sensor de Langmuir. Esta etapa del instrumento incluye un convertidor análogo digital (modelo AD7656 de Analog Devices), un microprocesador (modelo AT32UC3A3256 UC3-A3 Xplained de Microchip Technology) y una memoria (micro sd hc clase 10 con 32 Gb de Sandisk). En particular se presentan los requerimientos que justifican las decisiones de diseño y de implementación. Se describe la configuración usada dada los requerimientos y restricciones. También se presentan las pruebas realizadas a cada uno de los dispositivos para evaluar el correcto desempeño. En las pruebas fue posible corroborar que el sistema digital configurado puede operar a una tasa de muestreo de 14.41 k[Hz], detectando cambios de hasta 138,8 μ [s], a una tasa de consumo promedio de 0,297 [W]. Además, fue posible corroborar que la etapa digital funciona apropiadamente con el software de vuelo del SUCHAI-2 y -3.

Le dedico esta memoria a mi madre, Paulina Navarrete, y a mis abuelos, Nancy Gómez y Manuel Navarrete. Sin ellos, nada de esto habría sido posible. Además, agradezco a Carmen Navarrete y Marisol Sandoval, por su ayuda y motivación constante. También a mi polola, Magdalena Bassi, a mis tíos, mis primos y amigos (como al intrépido José Ogalde) <3. Todos ellos me alentaron en el día a día, me ayudaron a no dejarme desanimar por las adversidades que se presentan en la marcha... a tener esperanza en el futuro, a soñar muy alto.

Tabla de Contenido

Índice de Ilustraciones	viii
1. Introducción	1
1.1. Motivación y Antecedentes	1
1.2. Objetivo General	3
1.3. Objetivos Específicos	3
2. Revisión Bibliográfica y Estado del Arte	5
2.1. Introducción Histórica	5
2.2. Plasma Ionosférico	6
2.3. El instrumento Langmuir Probe	8
2.4. Problemas que presenta el Langmuir Probe	9
2.5. El Langmuir Probe que lleva el SUCHAI-I	11
2.6. Nuevo modelo de Langmuir Probe y ventajas	12
2.7. Objetivo de la Memoria	13
3. Metodología	15
3.1. El montaje Planar Langmuir Probe ya construido y el SBC Arduino Due (modelo del microprocesador: AT91SAM3X8E): Integrar, evaluar y calibrar: un ADC y un microprocesador.	15
3.2. El montaje Planar Langmuir Probe ya construido y el SBC AVR uc3-a3 Xplained (modelo del microprocesador: AT32UC3A3256). Integrar, evaluar y calibrar: un ADC y un microprocesador.	16
4. Implementación	18
4.1. El montaje de la tarjeta Planar Langmuir Probe ya construida y el SBC Arduino Due (modelo del microprocesador: AT91SAM3X8E).	18
4.2. El montaje de la tarjeta Planar Langmuir Probe ya construida y el AVR uc3-a3 Xplained (modelo del microprocesador: AT32UC3A3256).	23
4.2.1. Calibración en valores de voltajes constantes: generar gráficos y comparar el desempeño visto frente a la información del fabricante.	27
4.2.2. Primera calibración en frecuencia: organización del código y disminución de la tasa de sampleo de la conexión SPI.	29
4.2.3. Segunda calibración en frecuencia: implementación de una 'interrupción' y disminución de la tasa de sampleo de la conexión SPI.	31
5. Resultados	35

5.1.	El montaje de la tarjeta Planar Langmuir Probe ya construido y el SBC Arduino Due (modelo del microprocesador: AT91SAM3X8E).	35
5.1.1.	Calibración en valores de voltajes constantes: generar gráficos y comparar el desempeño frente al fabricante.	36
5.2.	El montaje de la tarjeta Planar Langmuir Probe ya construido y el SBC AVR uc3-a3 Xplained (modelo del microprocesador: AT32UC3A3256).	36
5.2.1.	Calibración en valores de voltajes constantes: generar gráficos y comparar el desempeño frente al fabricante.	38
5.2.2.	Primera calibración en frecuencia: organización del código y disminución de la tasa de muestreo de la conexión SPI.	39
5.2.3.	Segunda calibración en frecuencia: implementación de una 'interrupción' y disminución de la tasa de muestreo de la conexión SPI.	43
6.	Conclusiones	56
6.1.	Conclusiones	56
6.1.1.	Calibración en valores de voltajes constantes: generar gráficos y comparar el desempeño frente al fabricante.	56
6.1.2.	Primera calibración en frecuencia: organización del código y disminución de la tasa de muestreo de la conexión SPI.	57
6.1.3.	Segunda calibración en frecuencia: implementación de una 'interrupción' y disminución de la tasa de muestreo de la conexión SPI.	58
6.2.	Proposición de Trabajo a Futuro	60
7.	Bibliografía	62
	Anexos	64
A.		64
A.1.	ADC Parameters Unit Conversion.pdf	65
A.2.	Planar Langmuir Probe - Circuit Simulation - v1.pdf	74
A.3.	Planar Langmuir Probe - Measurement Sheet.pdf	87
B.		89
B.1.	Code: 'Main.c'	89
B.2.	Code: 'langmuirProbe.h'	92
B.3.	Code: 'langmuirProbe.c'	94
B.4.	Code: 'FIT_load_and_graf_all_test_freq_inter.m'	102
B.5.	Code: 'FIT_load_test_freq_interruption.m'	104

Índice de Ilustraciones

2.1.	<i>Perfiles típicos de las curvas de densidad de electrones (diurnos y nocturnos) que muestran las regiones D, E y F de la ionósfera con respecto a la altura en latitudes medias del planeta Tierra. También, en la figura se muestran las curvas para la máxima y mínima radiación solar. Además, en la figura se entregan perfiles para el día y para la noche [12].</i>	6
2.2.	<i>Perfiles típicos para la temperatura de neutrones, iones y electrones (diurnos y nocturnos) con respecto a la altura en latitudes medias del planeta Tierra. También, en la figura se muestran las curvas para la máxima y mínima radiación solar. Además, en la figura se entregan perfiles para el día y para la noche [12].</i>	7
2.3.	<i>(A la izquierda) Un circuito básico de la sonda Langmuir Probe basado en un amplificador de Trans-impedancia (TIA) utilizado como interfaz para un convertidor analógico a digital (ADC) [12].</i>	9
2.4.	<i>(a la derecha) la curva I-V que dicha sonda observaría al estar inmersa en un plasma [12].</i>	10
2.5.	<i>Se muestra la curva I-V del Langmuir Probe cuando el electrodo está limpio y sucio. El eje horizontal es el voltaje y el eje vertical es la corriente, en el captador de partículas. Además, se muestra la diferencia de las curvas al ser recorridas en distintos sentidos [24].</i>	11
2.6.	<i>(A la izquierda) Imagen de la tarjeta del sensor Langmuir desarrollada y fabricada en el laboratorio SPEL para el SUCHAI-I. (A la derecha) ubicación de componentes electrónicos en el esquemático del circuito diseñado [20]. . .</i>	12
2.7.	<i>Esquema de bloques del funcionamiento del sensor Langmuir Probe que fue diseñado en el laboratorio SPEL. A grandes rasgos se puede ver el electrodo, la parte analógica y la parte digital. Posterior a esto vendría la etapa de estimación de las variables de interés.</i>	14
4.1.	<i>Foto del primer montaje del circuito construido para cambiar el voltaje. Este circuito se implementó finalmente con éxito en una protoboard. Se le agregaron filtros pasa bajo (del tipo RC) a las entradas para eliminar el ruido.</i>	20
4.2.	<i>Foto de la fuente de alimentación que energiza la tarjeta Planar Langmuir Probe. Necesita ser alimentada con -17[V], 0[V] y 17[V]. Habitualmente consume 0.03[A] en la terminal negativa y 0.05[A] en la terminal positiva.</i>	21
4.3.	<i>Ejemplo de una de las gráficas que se logró con los datos obtenidos de la calibración del sistema ADC ocupando el microprocesador Arduino Due. Se muestra los valores de salida respecto a los valores de entrada.</i>	22

4.4.	<i>Esquema digital del diseño de la tarjeta que se terminó de montar. Esta es la conexión entre la tarjeta SBC y el resto del instrumento. El diseño y fabricación es de ingenieros del laboratorio SPEL (Gerardo Peña).</i>	24
4.5.	<i>La figura superior muestra el ciclo de trabajo del ADC, esto consiste en los estados que adquieren ciertos “pines” o terminales de interés. Este ciclo debe ser ejecutado para que el ADC funcione correctamente y pueda transformar las señales analógicas en lecturas digitales. Además, en la figura superior nos muestran los niveles de voltaje que adoptan los “pines” respecto al tiempo, a la izquierda se detallan los nombres de los “pines” del ADC. La figura inferior muestra el detalle de uno de los 6 mensajes correspondientes al pin “DATA”. La figura fue obtenida desde [16].</i>	25
4.6.	<i>La figura nos muestra la impresión que se realizó en el monitor serial CuteCom de los seis canales de datos leídos por la tarjeta AVR uc3-a3 Xplained desde el ADC.</i>	26
4.7.	<i>La figura nos muestra el diseño del circuito que se construyó, para entregar valores entre -15[V] a 15[V]. Se ocupó un potenciómetro, filtros y un OPAM para el “Amplificador Seguidor de Voltaje”.</i>	28
4.8.	<i>En la figura se muestra la grafica de la respuesta en frecuencia que alcanzan dos señales de salida de la parte analógica del Planar Langmuir Probe: “Low Gain” y “High Gain”. Estas señales representan las mediciones de voltaje del electrodo [20].</i>	29
5.1.	<i>La figura es un esquema que muestra el problema (desfase) de corrimiento que presentaban las impresiones por el monitor serial.</i>	36
5.2.	<i>Gráfica del comportamiento de los datos de salida del ADC (leídos con la tarjeta SBC Arduino Due) y de su regresión lineal, respecto a los datos de entrada leídos desde el osciloscopio. La abreviación ‘Mod. Offset’ indica el aumento en la precisión de los datos gracias al ‘Voltaje de Offset’. Se corrigieron los decimales de cada medida.</i>	37
5.3.	<i>Gráfica del Error Medio entre los datos de salida del ADC (leídos con la tarjeta SBC Arduino Due) y los datos de entrada al ADC (leídos por el osciloscopio), respecto a los datos de entrada al ADC. Se muestran dos curvas: la curva sin corregir y la curva corregida con los Voltajes Offset entregados por el osciloscopio.</i>	38
5.4.	<i>Gráfica de la diferencia entre los datos de salida del ADC (leídos con la tarjeta SBC Arduino Due) y su regresión lineal, respecto a los datos de entrada al ADC (leídos por el osciloscopio). Se muestran dos curvas: la curva sin corregir y la curva corregida con los Voltajes Offset entregados por el osciloscopio.</i>	39
5.5.	<i>Gráfica de las desviaciones estándar de los datos de entrada al ADC (leídos por el osciloscopio) y de los datos de salida del ADC (leídos con la tarjeta SBC Arduino Due), respecto a los datos de entrada al ADC. Todos los valores son positivos y su magnitud bordea los [mV].</i>	40
5.6.	<i>Gráfica del comportamiento de los datos de salida del ADC (leídos con la tarjeta SBC AVR uc3-a3 Xplained), respecto a los datos de entrada leídos desde el osciloscopio. Se pueden ver las dos zonas grandes de operación: la zona lineal ($V_{out} \leq 10 [V]$), en donde tiene un comportamiento afín ($f(x) \simeq x$) y la zona saturada, en los extremos donde se aplana la curva ($V_{out} \geq 10 [V]$).</i>	41

5.7.	<i>Gráfica del comportamiento de los datos de salida del ADC (leídos con la tarjeta SBC AVR uc3-a3 Xplained) y de su regresión lineal, respecto a los datos de entrada leídos desde el osciloscopio. Se corrigieron los decimales de cada medida gracias al 'Voltaje de Offset' entregados por el osciloscopio. . . .</i>	42
5.8.	<i>Gráfica del Error Medio entre los datos de salida del ADC (leídos con la tarjeta SBC AVR uc3-a3 Xplained) y los datos de entrada al ADC (leídos por el osciloscopio), respecto a los datos de entrada al ADC. Además, se graficó la regresión lineal de la curva. Se corrigieron los decimales de cada medida gracias al 'Voltaje de Offset' entregados por el osciloscopio.</i>	43
5.9.	<i>Gráfica de la diferencia entre los datos de salida del ADC (leídos con la tarjeta SBC AVR uc3-a3 Xplained) y la regresión lineal que se puede construir de los datos de salida del ADC, respecto a los datos de entrada al ADC (leídos por el osciloscopio). Además, se graficó el promedio de las mediciones. Se corrigieron los decimales de cada medida gracias al 'Voltaje de Offset' entregados por el osciloscopio.</i>	44
5.10.	<i>Gráfica de la curva de la desviación estándar de los datos de entrada al ADC (leídos por el osciloscopio) y grafica de la curva de la desviación estándar de los datos de salida del ADC (leídos con la tarjeta SBC AVR uc3-a3 Xplained), respecto a los datos de entrada al ADC. Todos los valores son positivos y su magnitud bordea los [mV]. Además, se graficó el promedio de cada curva de mediciones.</i>	45
5.11.	<i>Gráfica conjunta del comportamiento de la función "Fast Fourier Transform" calculada para las mediciones de las ocho señales de entrada al ADC (leídas desde el osciloscopio), respecto al espectro de frecuencia. Cada curva tiene un color distinto y se puede ver la componente fundamental en frecuencia de esa señal (0.08k[Hz], 0.2k[Hz], 0.5k[Hz], 1k[Hz], 3k[Hz], 5k[Hz], 7k[Hz] y 10k[Hz]).</i>	46
5.12.	<i>Gráfica conjunta del comportamiento de la función "Power Spectral Density" calculada para las mediciones de las ocho señales de entrada al ADC (leídas desde el osciloscopio), respecto al espectro de frecuencia. Cada curva tiene un color distinto y se puede ver la componente fundamental en frecuencia de esa señal (0.08k[Hz], 0.2k[Hz], 0.5k[Hz], 1k[Hz], 3k[Hz], 5k[Hz], 7k[Hz] y 10k[Hz]).</i>	47
5.13.	<i>Gráfica conjunta del comportamiento de la función "Fast Fourier Transform" calculada para las mediciones de las ocho señales de salida del ADC (leídas con la tarjeta SBC AVR uc3-a3 Xplained e impresas desde el monitor serial "CuteCom"). Cada curva tiene un color distinto y se puede ver la componente fundamental en frecuencia de esa señal (0.08k[Hz], 0.2k[Hz], 0.5k[Hz], 1k[Hz], 3k[Hz], 5k[Hz] y 7k[Hz]). Además, se puede apreciar la aparición de una nueva curva con una frecuencia fundamental en 7.45k[Hz] aproximadamente. Esta curva es el resultado de la distorsión de la curva con una frecuencia fundamental en 10k[Hz] respectivamente. Esto es debido a que posee una frecuencia fundamental más altas que la frecuencia máxima de calibración (8.7k[Hz]). .</i>	48

- 5.14. *Gráfica conjunta del comportamiento de la función “Power Spectral Density” calculada para las mediciones de las ocho señales de salida del ADC (leídas con la tarjeta SBC AVR uc3-a3 Xplained e impresas desde el monitor serial “CuteCom”). Cada curva tiene un color distinto y se puede ver la componente fundamental en frecuencia de esa señal (0.08k[Hz], 0.2k[Hz], 0.5k[Hz], 1k[Hz], 3k[Hz], 5k[Hz] y 7k[Hz]). Además, se puede apreciar la aparición de una nueva curva con una frecuencia fundamental en 7.45k[Hz] aproximadamente. Esta curva es el resultado de la distorsión de la curva con una frecuencia fundamental en 10k[Hz] respectivamente. Esto es debido a que posee una frecuencia fundamental más altas que la frecuencia máxima de calibración (8.7k[Hz]).* 49
- 5.15. *Gráfica conjunta del comportamiento de la “Función de Transferencia” calculada para las mediciones de las ocho señales de entrada y salida del ADC (leídas con el osciloscopio y la tarjeta SBC AVR uc3-a3 Xplained respectivamente). Cada curva tiene un color distinto y se puede ver la componente fundamental en frecuencia de esa señal (0.08k[Hz], 0.2k[Hz], 0.5k[Hz], 1k[Hz], 3k[Hz], 5k[Hz] y 7k[Hz]). Además, se puede apreciar la aparición de una nueva curva con una frecuencia fundamental en 7.45k[Hz] aproximadamente. Esta curva es el resultado de la alteración de la curva con una frecuencia fundamental en 10k[Hz] respectivamente. Esto es debido a que posee una frecuencia fundamental más altas que la frecuencia máxima de calibración (8.7k[Hz]). Por ultimo se realizó una interpolación (ajuste del tipo “cubicspline”) de las frecuencias fundamentales de todas las curvas que no sufrieron alteraciones de corrimiento en frecuencia.* 50
- 5.16. *Gráfica conjunta del comportamiento de la función “Fast Fourier Transform” calculada para las mediciones de las ocho señales de entrada al ADC (leídas desde el osciloscopio), respecto al espectro de frecuencia. Cada curva tiene un color distinto y se puede ver la componente fundamental en frecuencia de esa señal (0.125k[Hz], 0.25k[Hz], 0.5k[Hz], 1k[Hz], 2k[Hz], 4k[Hz], 8k[Hz] y 12k[Hz]).* 51
- 5.17. *Gráfica conjunta del comportamiento de la función “Power Spectral Density” calculada para las mediciones de las ocho señales de entrada al ADC (leídas desde el osciloscopio), respecto al espectro de frecuencia. Cada curva tiene un color distinto y se puede ver la componente fundamental en frecuencia de esa señal (0.125k[Hz], 0.25k[Hz], 0.5k[Hz], 1k[Hz], 2k[Hz], 4k[Hz], 8k[Hz] y 12k[Hz]).* 52
- 5.18. *Gráfica conjunta del comportamiento de la función “Fast Fourier Transform” calculada para las mediciones de las ocho señales de salida del ADC (leídas con la tarjeta SBC AVR uc3-a3 Xplained e impresas desde el monitor serial “CuteCom”). Cada curva tiene un color distinto y se puede ver la componente fundamental en frecuencia de esas señales (0.125k[Hz], 0.25k[Hz], 0.5k[Hz], 1k[Hz], 2k[Hz], 4k[Hz]). Además, se puede apreciar la aparición de dos nuevas curvas con frecuencias fundamentales en 2.45k[Hz] y en 6.45k[Hz] aproximadamente. Estas dos curvas son el resultado de la distorsión de las curvas con frecuencias fundamentales en 12k[Hz] y 8k[Hz] respectivamente. Esto es debido a que poseen frecuencias fundamentales más altas que la frecuencia máxima de calibración (6k[Hz]).* 53

- 5.19. *Gráfica conjunta del comportamiento de la función “Power Spectral Density” calculada para las mediciones de las ocho señales de salida del ADC (leídas con la tarjeta SBC AVR uc3-a3 Xplained e impresas desde el monitor serial “CuteCom”). Cada curva tiene un color distinto y se puede ver la componente fundamental en frecuencia de esas señales (0.125k[Hz], 0.25k[Hz], 0.5k[Hz], 1k[Hz], 2k[Hz], 4k[Hz]). Además, se puede apreciar la aparición de dos nuevas curvas con frecuencias fundamentales en 2.45k[Hz] y en 6.45k[Hz] aproximadamente. Estas dos curvas son el resultado de la distorsión de las curvas con frecuencias fundamentales en 12k[Hz] y 8k[Hz] respectivamente. Esto es debido a que poseen frecuencias fundamentales más altas que la frecuencia máxima de calibración (6k[Hz]).* 54
- 5.20. *Gráfica conjunta del comportamiento de la “Función de Transferencia” calculada para las mediciones de las ocho señales de entrada y salida del ADC (leídas con el osciloscopio y la tarjeta SBC AVR uc3-a3 Xplained respectivamente). Cada curva tiene un color distinto y se puede ver la componente fundamental en frecuencia de esas señales (0.125k[Hz], 0.25k[Hz], 0.5k[Hz], 1k[Hz], 2k[Hz], 4k[Hz]). Además, se puede apreciar la aparición de dos nuevas curvas con frecuencias fundamentales en 2.45k[Hz] y en 6.45k[Hz] aproximadamente. Estas dos curvas son el resultado de la distorsión de las curvas con frecuencias fundamentales en 12k[Hz] y 8k[Hz] respectivamente. Esto es debido a que poseen frecuencias fundamentales más altas que la frecuencia máxima de calibración (6k[Hz]). Por ultimo se realizó una interpolación (ajuste del tipo “cubicspline”) de las frecuencias fundamentales de todas las curvas que no sufrieron alteraciones de corrimiento en frecuencia.* 55

Capítulo 1

Introducción

1.1. Motivación y Antecedentes

'Satellite of the University of Chile for Aerospace Investigation' (SUCHAI-I, Satélite de la Universidad de Chile para la Investigación Aeroespacial) [9] es el primer CubeSat chileno desarrollado por estudiantes de pregrado, ingenieros y profesores de Ingeniería Eléctrica, Física e Ingeniería Mecánica del 'Space and Planetary Exploration Laboratory' (SPEL), el cual se ubica en el Departamentos de Ingeniería Eléctrica de la Facultad de Ciencias Físicas y Matemáticas (FCFM) de la Universidad de Chile (Santiago, Chile). Este es un nanosatélite de aproximadamente 1 litro de volumen (10cm x 10cm x 10cm) y de 1 kg de masa, construido bajo el estándar "Cubesat". El objetivo principal de este proyecto era estudiar el proceso de diseño, construcción, integración, lanzamiento y operación de un nano-satélite tipo Cubesat.

Los Cubesat son un estándar satelital que imponen restricciones a la forma, volumen, masa y distribución de la masa del satélite, lo que permite reducir los costos de lanzamiento [8]. El Cubesat posee una estandarización escalable definida en cubos de 10 cm de arista y masa inferior a 1,33 kg. La mayor parte del desarrollo y lanzamientos de este tipo de satélites proviene de entornos académicos [5]. El término Cubesat es acuñado para denotar nanosatélites que se adhieren a los estándares descritos en el documento de especificaciones de CubeSats [1].

La trayectoria del satélite chileno SUCHAI-I, orbita al planeta Tierra a una altura promedio aproximada de 505 kilómetros respecto a la superficie y a una velocidad de $7000[m/s]$. Viaja por la atmósfera sin estimación, ni control de orientación. La altura a la que viaja corresponde a la zona de la atmósfera comprendida como la Ionósfera.

La Ionósfera es una parte de la atmósfera terrestre en la que los átomos o moléculas que la conforman están divididos en partículas cargadas eléctricamente, estos son conocidos como iones. Esta ionización ocurre debido a la incidencia de fotones (provenientes del sol) en las partículas de la zona. Las características de la ionósfera, como la altitud, la temperatura y el grado de ionización (cantidad de partículas ionizadas respecto al total de partículas) no son homogéneas alrededor del planeta Tierra. Las características dependen del lugar (longitud

y latitud) y del tiempo (día y hora) en la que se observen y evalúen las mediciones. La trayectoria del satélite presenta un radio mayor que la altura de ionización máxima presente en la Ionósfera [17]. Los gases neutros en esta capa absorben la radiación solar de menor longitud de onda (rayos gamma y rayos X) que son altamente energéticos. La Ionósfera se sitúa entre la mesosfera y la exosfera, y en promedio se extiende aproximadamente entre los 60 km y 1000 km [2]. Si la radiación solar está presente (de día), las temperaturas en la ionosfera pueden llegar a 1.500 °K. Sin embargo, estas elevadas temperaturas no se corresponden con la sensación de calor que se observa en otras capas de la atmósfera, ya que en la ionósfera la densidad es considerablemente más baja. De noche la temperatura oscila alrededor de los 400 °K. La Ionósfera se modela como un fluido compuesto de partículas en estado de plasma.

Junto con sus otros sensores, el satélite SUCHAI-I alberga varios experimentos dedicados al estudio de física espacial. En específico para los intereses de este trabajo, nos centraremos en el instrumento Langmuir Probe, un sensor de plasma que toma mediciones del plasma característico a la Ionósfera. El SUCHAI-I fue una prueba de concepto para el sensor Langmuir, que permitió definir las características y requerimientos para misiones espaciales más sofisticadas como las ideadas para los satélites SUCHAI-II y SUCHAI-III [7]. El sensor Langmuir medirá las características del plasma de la ionósfera en varios puntos con relativa simultaneidad. Este sensor es uno de los más usados para el diagnóstico de plasma tanto en el espacio, como en experimentos de laboratorio [8]. El sensor Langmuir es capaz de detectar in situ las características eléctricas del medio en el que está inmerso. Este instrumento tiene cuatro partes principales: el captador de partículas (o electrodo), la etapa análoga, la etapa digital y la etapa de estimación de la densidad de partículas.

El captador de partículas es un detector en estado sólido, capaz de medir partículas de alta energía provenientes del espacio, partículas que atraviesan la Ionósfera. El captador de partículas consta de una superficie de metal, llamado electrodo, colocado en una región del satélite en donde el plasma incide. En el electrodo se establece un voltaje a voluntad que puede variar de forma preestablecida y conocida. Este voltaje atraerá partículas afines eléctricamente [20].

La etapa análoga es en donde la señal de entrada es modificada para luego poder ser procesada por la etapa digital. Primero un Amplificador de Trans-impedancia (TIA) convierte la pequeña corriente de entrada del electrodo en una salida de voltaje. Luego el voltaje se invierte, se desplaza a los valores positivos y se amplifica con un Amplificador Instrumental (IA). Luego se utiliza un filtro RC para eliminar señales de alta frecuencia con una frecuencia de corte de 16 kHz aproximadamente. Posteriormente la señal de voltaje se amplifica con un factor de 101, utilizando dos Amplificadores Operacionales (OPAMS) OPA227. En paralelo a lo descrito anteriormente, se incluyó 2 sensores de temperatura para medir este parámetro en 2 puntos de la placa del Langmuir [20].

La etapa digital es la etapa de digitalización, tratamiento y posterior almacenamiento de las señales. Por medio de un convertidor analógico a digital (ADC) AD7656, las señales son digitalizadas. Por medio de una conexión SPI, el microcontrolador es capaz de recibir las señales ahora completamente digitales. El microcontrolador depura las señales y posteriormente las envía por otra conexión SPI a una tarjeta microSD perteneciente a la misma tarjeta del instrumento Langmuir.

Finalmente tenemos la etapa de Estimación. Para poder hablar con seguridad de las variables que caracterizan el sistema que deseamos estudiar, es necesario observar muchos datos, detectar sus zonas de concentración y estimar hacia donde tiende su comportamiento. Con todo estos, podemos construir estimadores de las variables de importancia y concluir respecto a ciertas hipótesis.

El objetivo es generar una curva característica de corriente-voltaje (I-V). Si el voltaje de la sonda es negativo, los iones del plasma contenidos dentro de una cierta distancia se sentirán atraídos por el potencial del electrodo (debido a las características eléctricas del plasma). Si el voltaje de la sonda es positivo, los electrones son los que sentirán la atracción. La distancia de atracción es característica del electrodo escogido en el diseño del sensor. A partir de la curva característica de I-V del sensor Langmuir se puede obtener la densidad de los iones, la densidad de los electrones, y bajo ciertos regímenes de operación, la temperatura de los electrones [12].

1.2. Objetivo General

En vías de desarrollo del software y calibración del Backend digital del nuevo prototipo del payload “Langmuir Probe”, para que satisfaga los requerimientos del estándar Cubesat. El Backend consiste en la etapa de adquisición y procesamiento de datos que permite la interconexión de este payload a través del bus con el “On Board Computer” (OBC) del satélite SUCHAI.

1.3. Objetivos Específicos

Hay 3 grandes hitos a desarrollar:

1. Integrar, Evaluar y Programar el payload Planar Langmuir Probe controlado mediante una tarjeta Arduino Due. Esto implica obtener los datos desde el ADC y procesarlos.
2. Integrar, Evaluar y Programar el payload Planar Langmuir Probe controlado mediante una tarjeta AVR uc3-a3 Xplained. Esto implica obtener los datos desde el ADC y procesarlos.
3. Calibrar en valores constantes y en frecuencia el ADC (AD7656) para los valores provistos por la parte analógica del Langmuir Probe controlado mediante una tarjeta AVR uc3-a3 Xplained.

En cada uno de estos hitos se desarrollará alguna de las siguientes tareas:

1. Integrar y evaluar un ADC (AD7656).
 - Prototipado
 - Testeo del prototipado

- Correcciones

2. Integrar y evaluar un microprocesador.

- Prototipado
- Testeo del prototipado
- Correcciones

Capítulo 2

Revisión Bibliográfica y Estado del Arte

2.1. Introducción Histórica

El satélite SUCHAI I, lleva consigo un conjunto de unidades funcionales (Payloads), entre los cuales se encuentra un sensor de plasma ionosférico. Este instrumento, del tipo Langmuir Probe, fue diseñado en conjunto entre investigadores de la Universidad de Taylor (EE.UU.) e investigadores del Laboratorio de Exploración Espacial y Planetaria (SPEL, Chile). El instrumento mide las características del medio en el que viaja el satélite (medición in-situ). En particular, este sensor mide con un potencial fijo de carga positiva, y por lo tanto únicamente puede medir la densidad de carga de los electrones del plasma. Este tipo de polarización en el electrodo del sensor muestra problemas de generación de offset en la toma de muestras (corrimiento en el voltaje de referencia), debido al efecto de ‘Carga en Superficie’ en el satélite (el efecto de ‘Carga en Superficie’ lo explico más adelante).

Con el objetivo de mejorar la medición del instrumento se propuso la creación de un nuevo diseño de Langmuir. En esta nueva versión se ha implementado un electrodo con potencial fijo de carga negativa, que permite la medición de iones positivos del plasma ionosférico en el que estará inmerso. Este nuevo diseño del sensor, junto con un electrodo plano, fue concebido en colaboración entre investigadores del ‘Space and Atmospheric Instrumentation Laboratory’ (SAIL) de la Universidad Aeronáutica Embry Riddle (ERAU-Daytona Beach, Florida, USA) e investigadores del SPEL. El SAIL es dirigido por el PhD. Aroh Barjatya. El laboratorio SPEL es dirigido por el PhD. Marcos Díaz.

La Figura 2.6 nos muestra (a la izquierda) la tarjeta -diseñada, construida y casi completamente montada- de este último diseño del instrumento Planar Langmuir Probe. Se pueden ver a simple vista sus partes, como el receptáculo del electrodo, los componentes embebidos, los componentes pasivos y los puntos de prueba de señal. Además, (a la derecha) nos muestra una vista del esquemático del diseño digital en el programa de desarrollo de tarjetas de sistemas embebidos, EAGLE del desarrollador Autodesk [20].

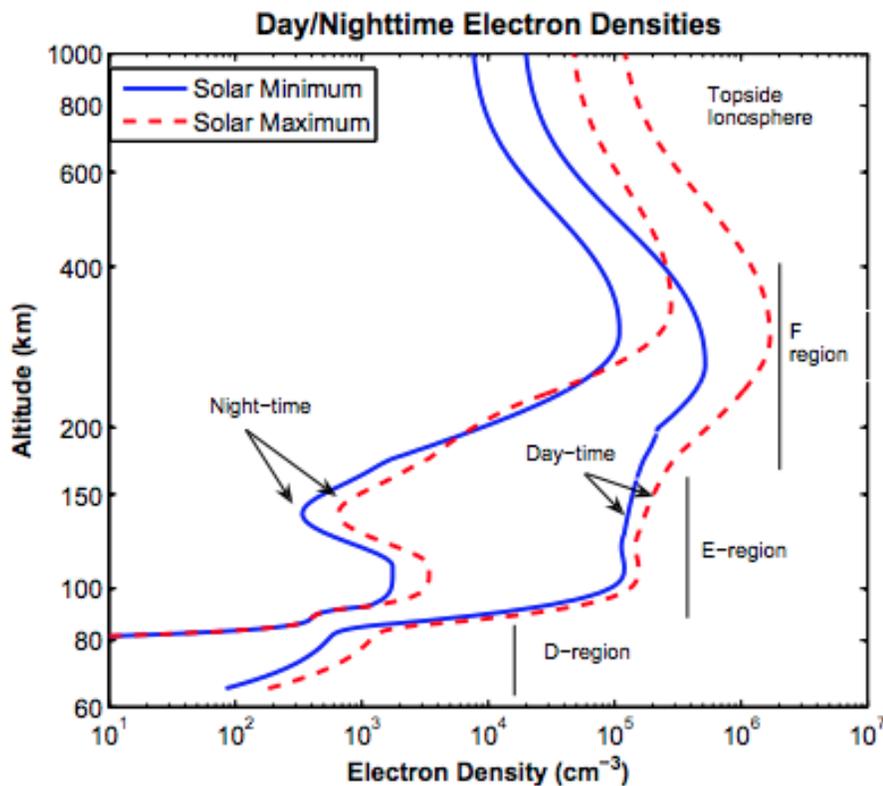


Figura 2.1: Perfiles típicos de las curvas de densidad de electrones (diurnos y nocturnos) que muestran las regiones D, E y F de la ionósfera con respecto a la altura en latitudes medias del planeta Tierra. También, en la figura se muestran las curvas para la máxima y mínima radiación solar. Además, en la figura se entregan perfiles para el día y para la noche [12].

2.2. Plasma Ionosférico

Al Plasma se le llama a veces 'el cuarto estado de la materia', además, de los tres conocidos: sólido, líquido y gaseoso. El plasma es un 'gas' en el que los átomos se han separado en partículas con carga (iones), y comúnmente está formado por electrones (negativos) y por iones (positivos). Estos últimos son átomos que han perdido electrones y han quedado con una carga eléctrica positiva y pueden moverse libremente [3].

El modelo "International Reference Ionosphere" (IRI), formado en los años 1960, es un proyecto internacional que tiene como objetivo desarrollar un modelo empírico estándar de la Ionósfera, basado en todas las fuentes de datos disponibles [4].

La ionósfera puede ser pensada como si estuviera compuesta por una serie de capas superpuestas en altitud (región D, E, F por nombrar algunas, en orden ascendente). Para cada capa se tiene una densidad máxima a una cierta altitud. Estas diversas regiones difieren en sus constituyentes de iones primarios y longitudes de onda de radiación Ultravioleta absorbidas. En la región D es predominante la aparición de iones hidratados, en la región E predominan los compuestos NO^+ y O_2^+ , y en la región F predomina el O^+ . La región D no existe por la noche, debido a la ausencia de ionización solar.

La figura 2.1 nos muestra el modelo IRI de densidad de electrones. Estos son los perfiles típicos de las curvas de densidad de electrones respecto a la altura en una posición de latitud media del planeta Tierra. Estas curvas varían de acuerdo con la radiación solar, en la figura se muestran las curvas para la máxima y mínima radiación solar recibida. Además, en la figura se detalla la posición de las capas superpuestas en altitud de la Ionosfera (región D, E y F). Además, en la figura se entrega un grafico de perfiles para el día y uno para la noche.

La figura 2.2 nos muestra el modelo IRI de temperatura para los iones (T_i), electrones (T_e) y neutrones (T_n). Estos son los perfiles típicos de las curvas de temperatura de esas partículas respecto a la altura en una posición de latitud media del planeta Tierra. Estas curvas varían de acuerdo con la radiación solar, en la figura se muestran las curvas para la máxima y mínima radiación solar recibida. Además, en la figura se entrega un grafico de perfiles para el día y uno para la noche.

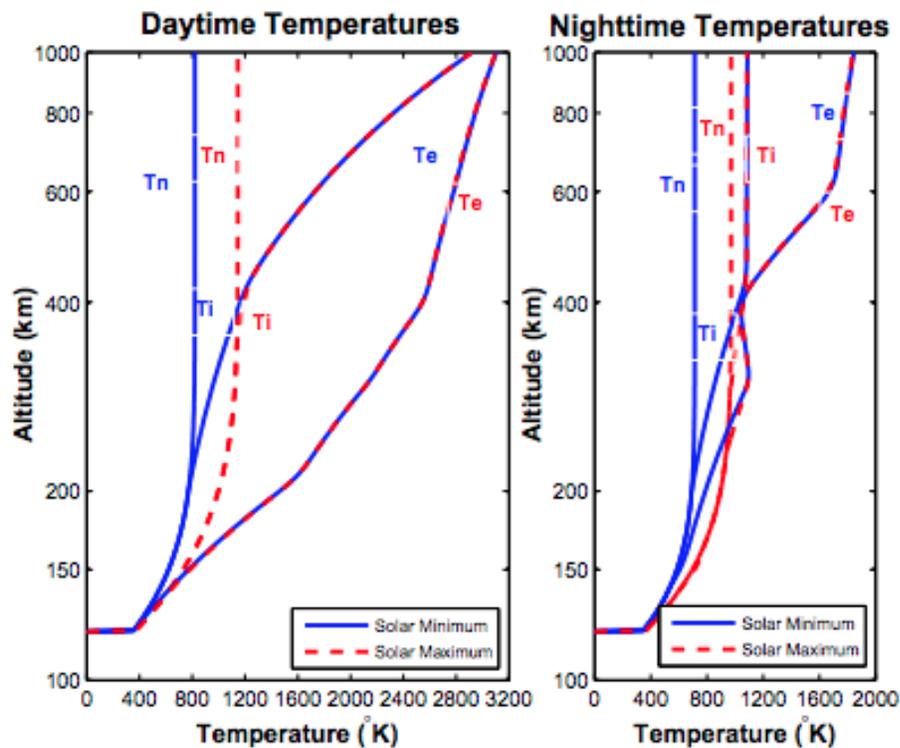


Figura 2.2: *Perfiles típicos para la temperatura de neutrones, iones y electrones (diurnos y nocturnos) con respecto a la altura en latitudes medias del planeta Tierra. También, en la figura se muestran las curvas para la máxima y mínima radiación solar. Además, en la figura se entregan perfiles para el día y para la noche [12].*

Los perfiles de temperatura y densidad ionosférica difieren con la latitud, la longitud y la estación, debido a la variación en el ángulo cenital de la radiación Ultravioleta incidente y del campo geomagnético de la Tierra [12]. El modelo ionosférico del proyecto IRI propone promedios climatológicos parametrizados para diferentes niveles de: actividad solar, estaciones e índices de actividad geomagnéticas. La variabilidad en tiempo real (in-situ) de la ionósfera en condiciones geomagnéticamente tranquilas puede acercarse en el mejor de los escenarios

a el 30% de los promedios proporcionados por el modelo, y comienza a equivocarse para condiciones geomagnéticamente alteradas [12] [14].

Entonces, los instrumentos en tiempo real son importantes para corregir el modelo ionosférico y para obtener datos con una resolución espacial y temporal alta. Esas mediciones son fundamentales para el estudio de los fenómenos de la ionosfera. Las sondas eléctricas fueron los primeros instrumentos en tiempo real para la medición de la densidad del plasma en la ionósfera. Una parte importante de las sondas eléctricas son generalizadas con el nombre de Langmuir Probe [12].

2.3. El instrumento Langmuir Probe

Una sonda Langmuir es un dispositivo utilizado para determinar propiedades del plasma como: la temperatura de los electrones, la densidad de los electrones y el potencial eléctrico del plasma. Funciona insertando uno o más electrodos en este medio, con un potencial eléctrico constante o variable en el tiempo, entre los diversos electrodos o entre ellos y el resto del instrumento circundante, y observando la corriente que se acumula. Las corrientes y potenciales medidos en este sistema permiten la determinación de las propiedades físicas del plasma. La corriente total observada I , es la suma de varias corrientes particulares: corrientes térmicas de electrones y iones, corriente de fotoelectrones, corrientes de electrones secundarios, etc. Sin embargo, las ventajas de la simplicidad de esta técnica se ven opacadas por la complejidad de la teoría requerida para analizar las curvas I-V que caracterizan un instante.

La Figura 2.3 nos muestra un diagrama del circuito típico de una sonda Langmuir. Primero un Amplificador de Trans-impedancia (TIA) convierte la pequeña corriente de entrada, proveniente de un electrodo, en una salida de voltaje. Luego, el voltaje se invierte, se desplaza a los valores positivos y se amplifica con un Amplificador Instrumental (IA).

La Figura 2.4 nos muestra la respuesta típica de la curva I-V de un instrumento Langmuir inmerso en un plasma con una distribución de velocidad Maxwelliana. La “Corriente Recolectada” en las “Regiones de Saturación” esta en función de la “Densidad del Plasma”, y la “Corriente Recolectada” en la “Región de Retardo” depende de la “Temperatura de los Electrones”. Como se muestra en la figura, el “Potencial Flotante” (V_f) del Payload (o satélite) es el potencial de equilibrio alcanzado por un cuerpo conductor sumergido en un plasma, de modo que la “Corriente Recolectada” total de los electrones e iones en el electrodo es cero. El “Potencial de Plasma” (V_p) es el potencial en el que no existen campo/envoltura electromagnética entre la superficie del electrodo y el plasma circundante. Un potencial inferior a V_p repele electrones, un potencial superior a V_p repele iones. En las expresiones teóricas de la “Corriente Recolectada”, la respuesta plasmática es relativa a V_p . Para un cuerpo conductor sumergido en plasma térmico, V_f es menor con respecto a V_p , como se muestra en la figura [12].

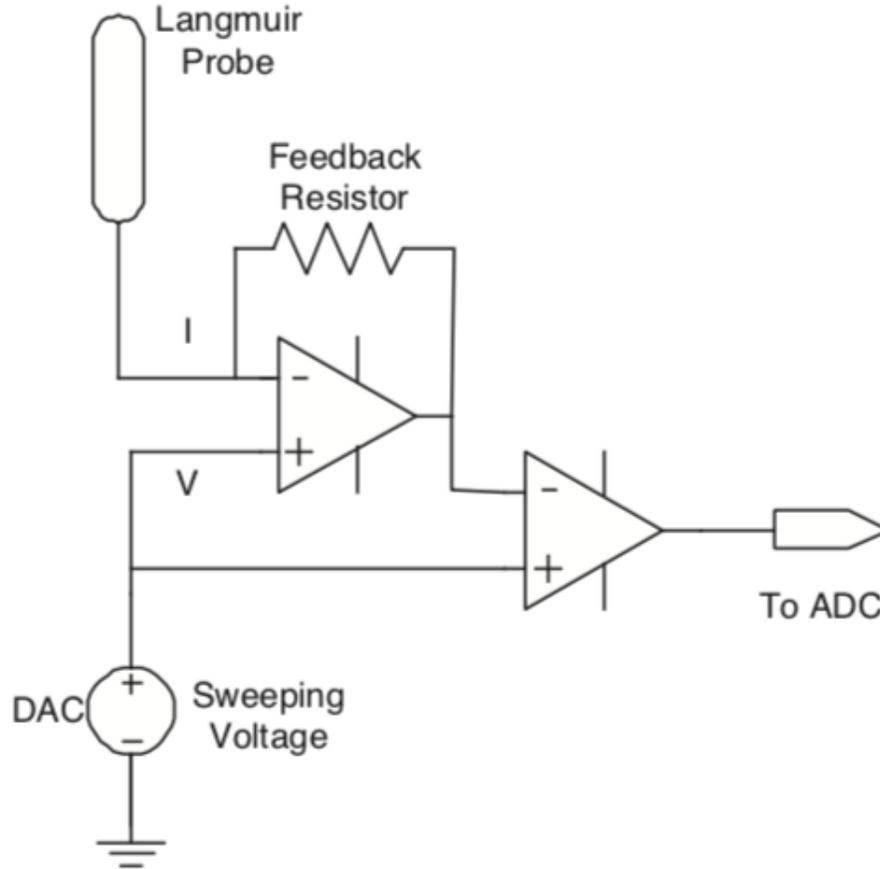


Figura 2.3: (A la izquierda) Un circuito básico de la sonda Langmuir Probe basado en un amplificador de Trans-impedancia (TIA) utilizado como interfaz para un convertidor analógico a digital (ADC) [12].

2.4. Problemas que presenta el Langmuir Probe

Este sensor evidencia problemas al estar presente en un satélite, estos tienen relación con su desempeño al momento de obtener mediciones, y son intrínsecos a su construcción y a su funcionamiento. Es importante conocer su existencia, ya que así se pueden corregir (hasta cierto punto) en las etapas posteriores del instrumento:

1. Contaminación en Superficie (Surface Contamination):

Sea cual sea el electrodo (capturador de partículas de estado sólido con el cual se realiza la medición) que se esté utilizando al momento de volar por el medio espacial, junto al resto del satélite, colisionan con partículas de distintos elementos que se adhieren a la superficie, y por lo tanto, “ensucian” el exterior de la nave. Al aumentar el tiempo de vuelo este efecto incrementa el error de las mediciones del sensor. Cambia la curva I-V característica [19].

La Figura 2.5 nos muestra la curva I-V en dos situaciones, con el electrodo limpio y con el electrodo con partículas de distintos elementos adheridas. También se ve que el

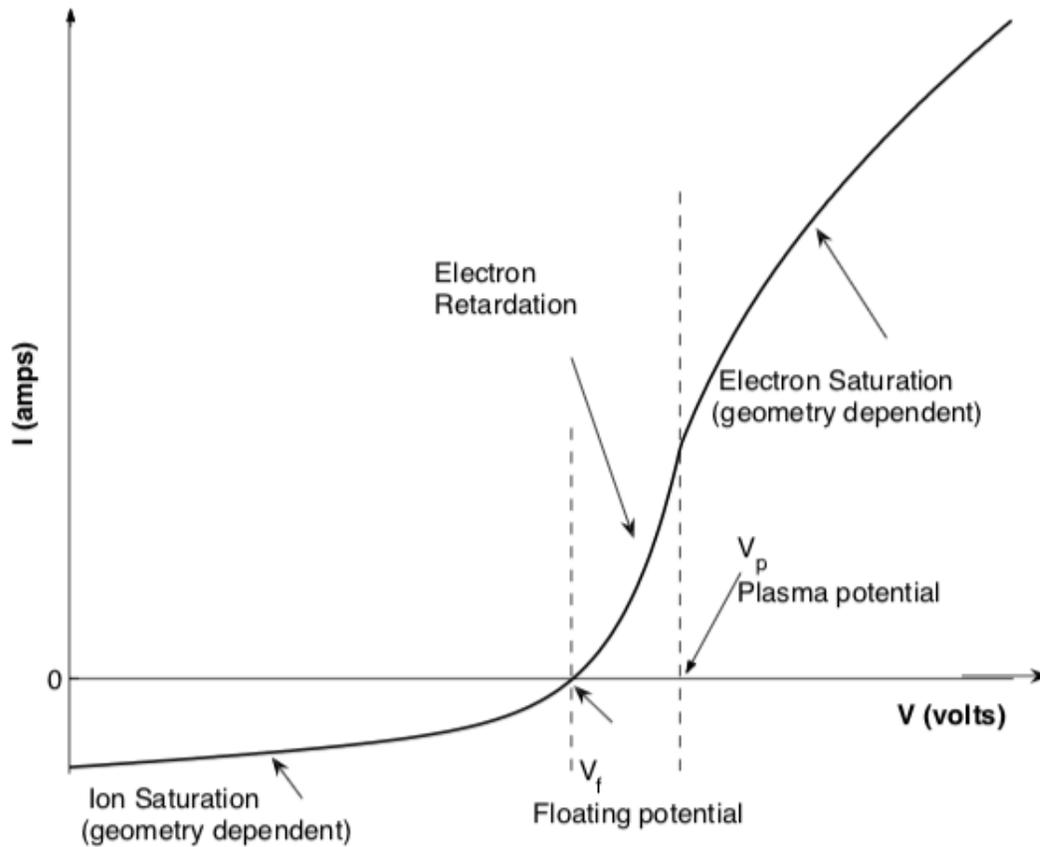


Figura 2.4: (a la derecha) la curva I - V que dicha sonda observaría al estar inmersa en un plasma [12].

comportamiento del registro de la curva cambia al ser recorrida la abscisa (voltaje en forma ascendente, luego de forma descendente). En el caso limpio la curva es la misma en los dos sentidos. En el caso sucio la curva recorre distintos lugares en cada sentido.

2. Carga en Superficie (Surface Charge):

Es un problema que aparece en los satélites de pequeño tamaño cuando ya llevan un tiempo en órbita. Este ocurre, debido a que, al llevar un potencial expuesto, las cargas opuestas (presentes en el plasma) son atraídas al satélite, se adhieren, y modifican la referencia a tierra del satélite (el voltaje nulo). Idealmente, se requiere una relación entre el área de superficie del satélite y el electrodo de 10.000 (o mayor), para eliminar este efecto. Esta proporción no se cumple para los nanosatélites, por lo tanto se debe encontrar un diseño, o tratamiento posterior de las medidas, que disminuya este problema [26].

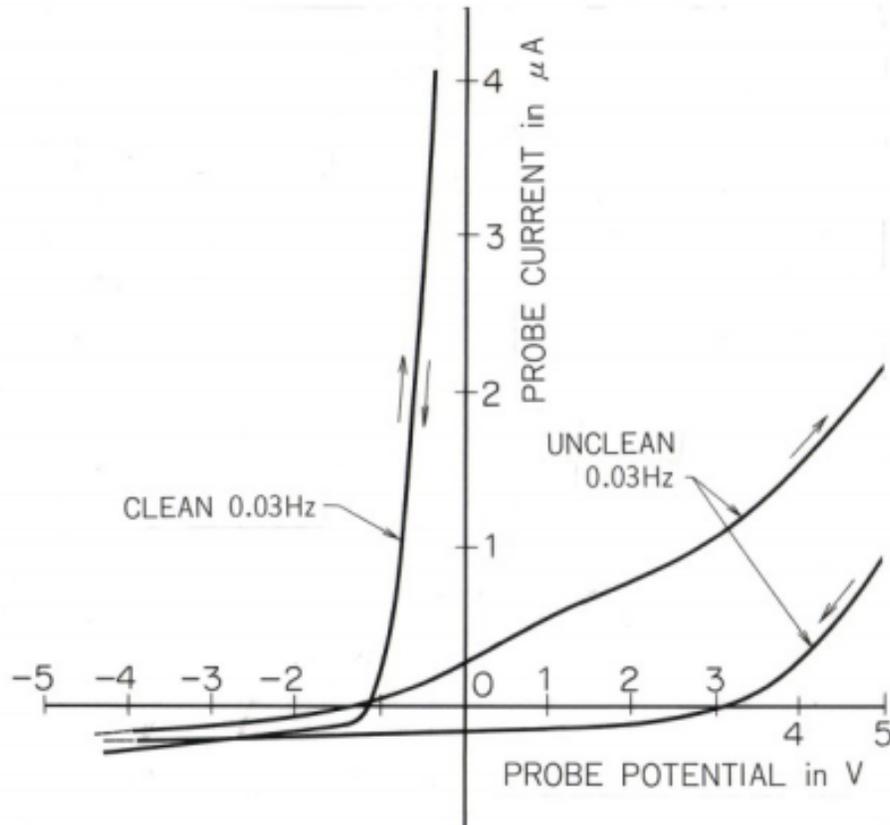


Figura 2.5: Se muestra la curva I-V del Langmuir Probe cuando el electrodo está limpio y sucio. El eje horizontal es el voltaje y el eje vertical es la corriente, en el captador de partículas. Además, se muestra la diferencia de las curvas al ser recorridas en distintos sentidos [24].

2.5. El Langmuir Probe que lleva el SUCHAI-I

El Langmuir Probe elegido para ir en el satélite SUCHAI-I fue diseñado por investigadores de Taylor University (Indiana, USA) y ya fue probado en el espacio en el satélite TSAT Cubesat, el cual fue lanzado el verano del año 2013. El satélite TSAT Cubesat llevaba consigo un sensor Langmuir Probe de barrido positivo que mide corrientes en escala logarítmica gracias al uso de un Electrómetro Logarítmico [15].

En el SUCHAI-I las mediciones son hechas usando un potencial de 4 voltios positivo fijo (Fixed Langmuir Probe), para obtener mediciones de la densidad de electrones. El electrodo que se utilizó fue uno con forma esférica y necesitaba poder salir al exterior del satélite.

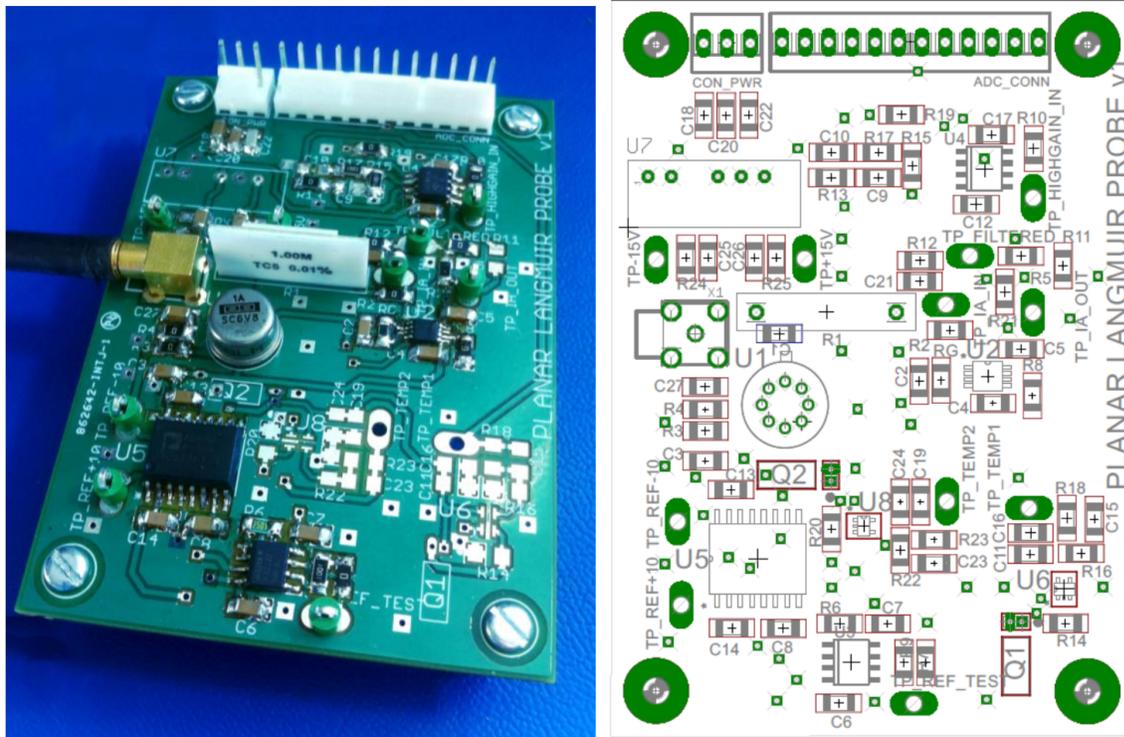


Figura 2.6: (A la izquierda) Imagen de la tarjeta del sensor Langmuir desarrollada y fabricada en el laboratorio SPEL para el SUCHAI-I. (A la derecha) ubicación de componentes electrónicos en el esquemático del circuito diseñado [20].

2.6. Nuevo modelo de Langmuir Probe y ventajas

Un nuevo diseño del instrumento Fixed Langmuir Probe se está desarrollando actualmente para los satélites SUCHAI-II y SUCHAI-III. El objetivo final a largo plazo, es lograr posicionar todos los elementos del payload (parte analógica, parte digital y la conexión con el resto del satélite) dentro de un nuevo diseño de la tarjeta del instrumento Langmuir. En la cual se desea reordenando sus componentes embebidos para que la tarjeta ocupe el menor espacio posible.

Respecto al hardware del payload, una característica de importancia es su electrodo plano (planar probe), que estará posicionado en la superficie del satélite que apunta en la orientación RAM (dirección y sentido en el que se mueve el satélite). Para esto es necesario que el satélite sea capaz de tener una estimación y control de orientación en todo momento. Además, el electrodo será polarizado en otro semiespacio de voltaje. Ya no será un valor positivo, si no que será polarizado con un potencial fijo negativo en la región de la saturación de los iones. Un instrumento Langmuir con estas características es conocido como “Planar Ion Probe” (PIP). El Planar Ion Probe mide las densidades de iones. Este diseño se ve afectado en menor medida por el efecto de Contaminación en Superficie [24] [25] y por el efecto de Carga en Superficie.[24] [26].

Respecto al software del payload, se desea que este payload sea controlado mediante el

software Suchai Flight Software. Este sistema operativo fue pensado para controlar un nanosatellite o varios de ellos, en específico el On Board Computer o las Single Board Computer de los payload. Fue diseñado por ingenieros e investigadores del laboratorio SPEL.

2.7. Objetivo de la Memoria

Ya fue construido el nuevo diseño del Planar Langmuir Probe en el laboratorio SPEL, sin embargo, éste presenta problemas de funcionamientos y de diseño. Esta placa Planar Langmuir Probe fue pensada para ser controlada a través de una “Single Board Computer” (SBC) del tipo Arduino DUE. Sin embargo, tampoco se logró obtener mediciones con este software.

Por lo tanto, en esta memoria se tendrá como primeras tareas: programar, testear y calibrar el software del SBC del tipo Arduino DUE, para así conseguir mediciones confiables provenientes de la parte análoga. Posteriormente se tendrá que: programar, testear y calibrar un nuevo software basado en el “SUCHAI Flight Software” (SFS) para manejar el Planar Langmuir Probe desde un SBC del tipo “AVR32-uc3 Xplained”. Con este software final se pondrá énfasis en aumentar la tasa de muestreo de los datos.

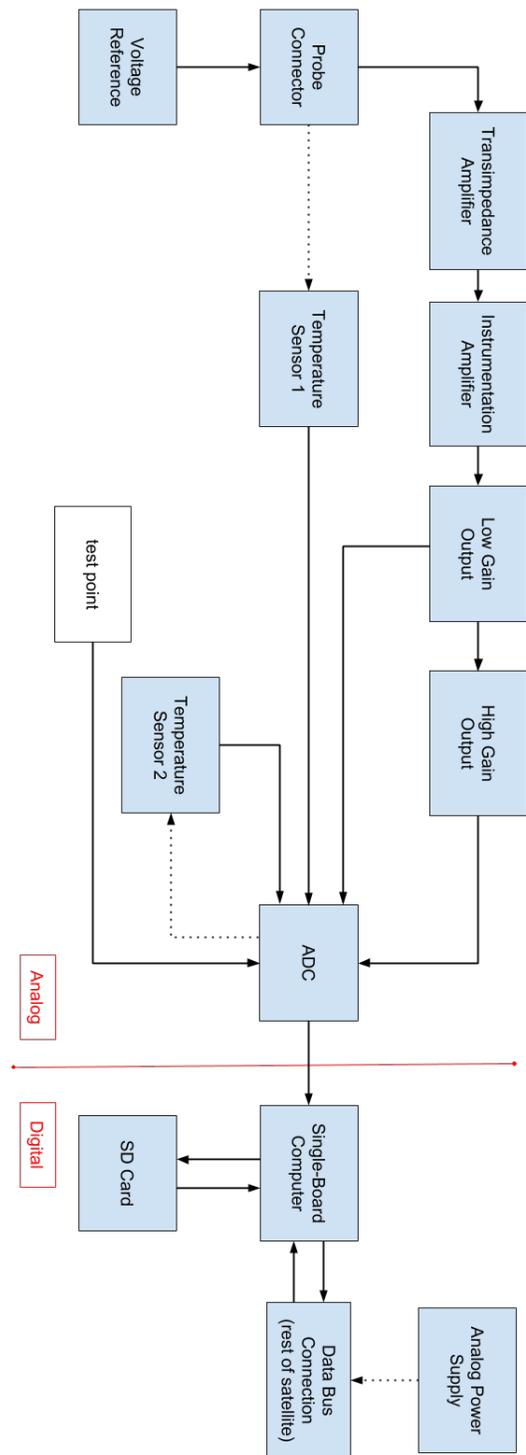


Figura 2.7: Esquema de bloques del funcionamiento del sensor Langmuir Probe que fue diseñado en el laboratorio SPEL. A grandes rasgos se puede ver el electrodo, la parte analógica y la parte digital. Posterior a esto vendría la etapa de estimación de las variables de interés.

Capítulo 3

Metodología

Para llevar a cabo esta memoria se deberán realizar 2 grandes hitos, los cuales corresponden a dos montajes distintos. El primer hito es trabajar con el montaje del Single Board Computer del tipo Arduino DUE junto con la tarjeta del Planar Langmuir Probe y modificar el software existente para manejar el sensor, conseguir sus mediciones y calibrar su funcionamiento. El segundo hito es trabajar con el montaje del SBC del tipo AVR32-uc3 Xplained junto con la tarjeta del Planar Langmuir Probe y desarrollar un software basado en el “SU-CHAI Flight Software” (SFS) para manejar el sensor, conseguir sus mediciones y calibrar su funcionamiento:

3.1. El montaje Planar Langmuir Probe ya construido y el SBC Arduino Due (modelo del microprocesador: AT91SAM3X8E): Integrar, evaluar y calibrar: un ADC y un microprocesador.

Ya existe un prototipo físico de la tarjeta del Planar Langmuir Probe y de la unidad de digitalización de sus mediciones, por lo tanto, las etapas de diseño y construcción de este prototipo fueron obviadas en esta memoria. Se trabajó directamente en lograr que todas las conexiones del prototipo físico fuesen las del diseño virtual (reparando las soldaduras y evitando cortos circuitos y circuitos abiertos), se trabajó en la modificación del software que lo controlaba y posteriormente se trabajó en la calibración de todo el montaje. El montaje no estaba funcionando correctamente, por lo tanto, se intentó entender cuáles eran sus problemas, y con esto, comprender su funcionamiento y solucionar los errores:

1. Montaje de esta configuración: Se utilizó: la tarjeta Planar Langmuir Probe, la fuente de poder externa para alimentar la tarjeta del PLP, la tarjeta SBC Arduino Due, el primer puerto SPI de la tarjeta Arduino Due para conectarla con la tarjeta PLP y una computadora. Las lecturas de los datos se hicieron en esta computadora junto a el programa “Arduino IDE” (Arduino Integrated Development Environment) a través de

la pantalla del monitor serial que trae integrado.

2. Integrar, evaluar y calibrar un ADC: Para ello se creó un circuito que fuese capaz de variar el voltaje entre -10 y 10 Voltios, en un punto determinado. Este circuito sirve para entregar distintos valores de voltaje al ADC, a través de un canal de testeo que posee el ADC. Con cada uno de estos valores de entrada conocidos, se obtendrá la respuesta del ADC, con lo cual se evaluará su funcionamiento. Teniendo estos datos, se desarrolló un código en el programa “Matlab” para comparar los resultados obtenidos respecto a los esperados. Además, se calcularon los errores asociados. A partir de esto, se realizaron correcciones al prototipo y se documentó con detalle cómo realizar estas pruebas tratando de minimizar los errores observados para el próximo prototipo a construir del instrumento Langmuir Probe.

3.2. El montaje Planar Langmuir Probe ya construido y el SBC AVR uc3-a3 Xplained (modelo del microprocesador: AT32UC3A3256). Integrar, evaluar y calibrar: un ADC y un microprocesador.

Como primera aproximación al sensor final, se desea poder controlar la tarjeta del Planar Langmuir Probe existente, con una tarjeta SBC del tipo AVR uc3-a3 Xplained, la cual trae incorporado un microprocesador modelo AT32UC3A3256. Esta tarjeta posee otra arquitectura, por lo tanto, las labores de integración y programación deberán ser desarrolladas por completo desde cero.

1. Montaje: Se desarrolló y trabajó un montaje que consiste en: el Planar Langmuir Probe, la fuente de poder externa para su alimentación, la tarjeta del SBC AVR uc3-a3 Xplained, una placa de conexión FTDI (que recibe la comunicación USART del microcontrolador y la entrega por USB al computador) y una computadora con la que se programó el software basado en el SFS que controla al SBC y procesa los datos.
2. Programación: Para esta configuración (tarjeta Langmuir y microprocesador) se debe programar un sistema capaz de comunicarse a la mayor velocidad posible de muestreo (interrumpiendo el ciclo habitual del microprocesador) a través del primer bus SPI en una dirección: desde el ADC al microcontrolador. Desde el ADC se enviarán las mediciones al microprocesador, en formato binario de 16 bits. La recepción de los datos por parte del software se desea realizar dentro de una interrupción de tiempo del tipo “Timer/Counter”. La cual permite aumentar la frecuencia de muestreo, para así conseguir lecturas digitales, con la mayor tasa de muestreo posible.
3. Calibración en valores de voltajes constantes: Se deberá realizar una calibración del instrumento Planar Langmuir Probe. Para esto se deben realizar pruebas de medición, con las que se obtendrán dos sets de datos. Valores de voltajes análogos constantes de entrada al ADC, leídos con un osciloscopio, y valores digitales de salida del ADC, leídos con la tarjeta del microprocesador. Con la comparación de estos dos sets de datos se podrá

caracterizar el comportamiento del instrumento a lo largo de toda su zona de operación en valores de voltaje constante. Los resultados de estas pruebas se compararán con los modelos de funcionamiento entregados por el fabricante y se evaluará el desempeño. Se requiere, además, un circuito capaz de entregar señales análogas de voltaje, las cuales serán los valores de entrada al ADC dentro de su rango de funcionamiento.

4. Calibración en valores de voltaje con componentes en frecuencia: También se deberá realizar una calibración en frecuencia del instrumento Planar Langmuir Probe. Para esto se deben realizar pruebas de medición, con las que se obtendrán dos sets de datos. Primero valores de voltajes análogos sinusoidales en la entrada al ADC (con distintas frecuencias fundamentales), leídos con un osciloscopio. También es necesario mencionar que estos valores de entrada serán espaciados a lo largo del rango de frecuencia, con el fin de ser posicionados dentro del rango máximo de frecuencia característico de las señales que entrega la parte analógica del instrumento Langmuir. Además, se obtendrán valores digitales de salida del ADC, leídos con la tarjeta del microprocesador. Con la comparación de estos dos sets de datos se podrá caracterizar el comportamiento del instrumento a lo largo de toda su zona de operación en el espectro de frecuencia. Se requiere, además, un generador de señales para los valores de entrada al ADC dentro de su rango de funcionamiento.

Capítulo 4

Implementación

Para llevar a cabo lo descrito en la metodología se deberá realizar una puesta en marcha, la cual consta de 2 grandes hitos. Éstos corresponden a 2 montajes distintos.

El primer hito es trabajar con el SBC Arduino DUE y el instrumento Planar Langmuir Probe. Las labores a realizar son: modificar el software existente para manejar el electrodo y la parte análoga, conseguir las mediciones y calibrar su funcionamiento. Se necesitará, además, un osciloscopio (para medir valores de voltajes análogos constantes), un circuito que pueda entregar esos voltajes en un punto con bajo ruido.

El segundo hito es trabajar con el SBC AVR32-uc3 Xplained y el instrumento Planar Langmuir Probe. Las labores a realizar son: desarrollar un software basado en el “SUCHAI Flight Software” (SFS) para manejar el electrodo y la parte análoga, conseguir las mediciones y calibrar su funcionamiento.

4.1. El montaje de la tarjeta Planar Langmuir Probe ya construida y el SBC Arduino Due (modelo del microprocesador: AT91SAM3X8E).

Se instaló en la tarjeta del SBC un código para el software Arduino IDE diseñado por colaboradores del laboratorio SPEL (Gerardo Peña Álvarez). Este código se programó y modificó para ser capaz de leer los datos enviados desde el ADC a través del primer bus SPI (Serial Peripheral Interface), en particular el “SPI0”. Este código fue programado en el lenguaje que utiliza el software “Arduino IDE” (Arduino Integrated Development Environment).

Para que la conexión SPI funcione correctamente, a partir de una configuración dada, se modificaron los siguientes parámetros: el modo ‘SPI_MODE3’ (que configura la polaridad, la fase y el borde de la señal del clock), el orden “MSBFIRST” (que configura al primer bit como el más significativo) y el divisor del clock “SPI_CLOCK_DIV128” (Establece al reloj que utilizará el SPI como la razón entre el reloj del software y 128).

Se imprimió por el monitor serial integrado del software “Arduino IDE”, los datos recibidos por la tarjeta del microcontrolador, provenientes desde los seis canales de recepción/emisión del ADC. Estos datos recibidos en la tarjeta SBC, enviados por el ADC e impresos en el monitor serial corresponden a: en los 4 primeros canales, al voltaje impuesto por el lado analógico del Planar Langmuir Probe, el quinto, al voltaje conectado a un punto de prueba llamado “TP_ADC_CH5” según el esquemático de la placa [21] y el sexto no se lee (no fue habilitado con una conexión, quedó en circuito abierto).

Es oportuno mencionar que por el mismo cable Universal Serial Bus (USB) con el que se carga y graba el software a la tarjeta SBC Arduino DUE, se recibe la transmisión de datos por parte del ordenador, que luego se visualiza en el monitor serial. En otros entornos de trabajo (controladores) se necesitan distintas conexiones USB para realizar estas tareas.

Debido a que el voltaje configurado en los pines DB10, DB11 y DB12 (del ADC) fue 0[V], el rango de entrada de voltaje analógico se limita a estar entre -10[V] y 10[V]. Se deseaba trabajar en este rango de voltaje, ya que el ADC es capaz de leer en este rango en cada uno de sus canales, esto debido a la configuración ajustada en el ADC. Dentro de este rango de valores el ADC presenta un comportamiento lineal. Fuera de estos valores el ADC presenta un comportamiento de saturación.

Se construyó un circuito capaz de entregar voltajes constantes en el rango de -10[V] a 10[V]. Respecto a la corriente entregada por este circuito no se aplicaron restricciones. Este circuito se implementó finalmente con éxito en una protoboard, y se le agregaron filtros pasa bajo (del tipo RC) a las entradas.

La Figura 4.1 nos muestra un primer intento de construcción del circuito capaz de entregar voltajes constantes. Se ocupó un potenciómetro en el centro para variar el valor de la resistencia, junto con resistencias constantes en las entradas para disminuir los voltajes y capacitores para disminuir el rizado de la señal de salida. La salida debía fluctuar entre -10[V] y 10[V].

La fuente de alimentación que se utilizó para el circuito fueron 3 puntos de la tarjeta Planar Langmuir Probe que entregan voltajes fijos y con bajo ruido. El nombre de estos puntos son “TP_V+”, “TP_V-”, y “TP_GND” de acuerdo con su diseño [21] y entregan los siguientes voltajes en el mismo orden: +15[V], -15[V] y 0[V]. El Planar Langmuir Probe se energizó utilizando una fuente de alimentación DC: GP-4303TP.

La Figura 4.2 nos muestra la fuente que se utilizó para energiza la tarjeta Planar Langmuir Probe. Necesita ser alimentada con 3 voltajes constantes en: -17[V], 0[V] y 17[V]. Habitualmente la corriente que consume es de 0.03[A] en la terminal negativa y de 0.05[A] en la terminal positiva. Posteriormente, con este circuito se recorrieron 22 valores equiespaciados en el rango entre -11[V] y 11[V]. Son 22 medidas equiespaciadas ya que así se puede recorrer este rango con una separación entre valores de ± 1 [V], lo cual nos entrega una discretización razonable del espacio de funcionamiento. Además, se quiso obtener mediciones en la zona de operación lineal, y otros, en la zona de operación saturada.

Se midió con la sonda análoga del osciloscopio modelo “Tektronix MDO4054B-3” cada uno de estos valores, los cuales provienen de los voltajes impuestos, por el circuito antes

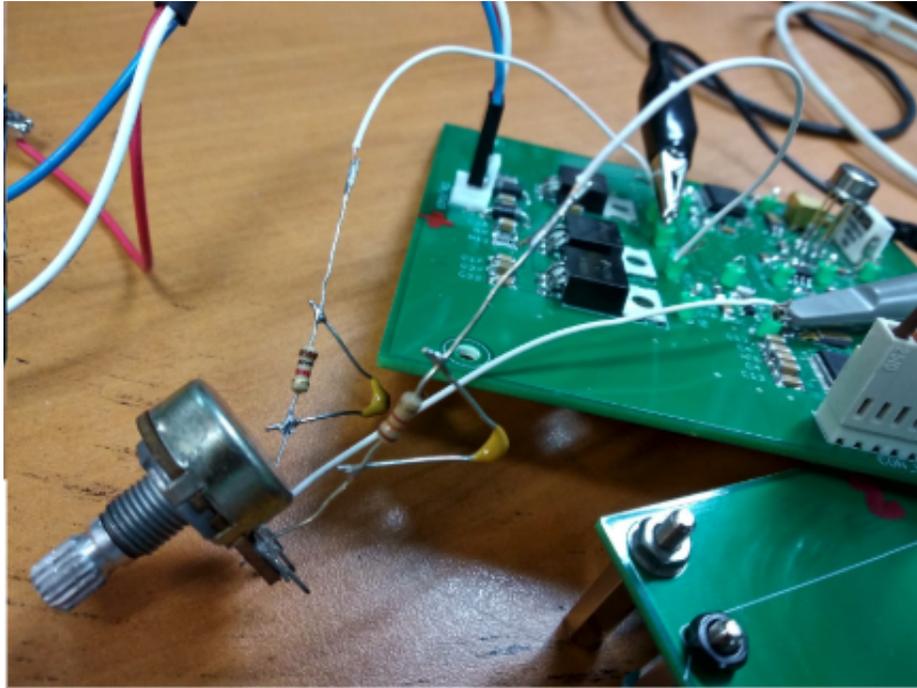


Figura 4.1: Foto del primer montaje del circuito construido para cambiar el voltaje. Este circuito se implementó finalmente con éxito en una protoboard. Se le agregaron filtros pasa bajo (del tipo RC) a las entradas para eliminar el ruido.

mencionado, en el canal “TP_ADC_CH5” de la tarjeta PLP. Se ocupó un osciloscopio, debido a que había que tener exactitud en los valores de entrada al sistema (con sistema comprendemos al embebido ADC). Cada una de las mediciones del osciloscopio guardadas corresponden a una colección de datos en una ventana de tiempo. Estos datos tienen una precisión mayor o igual a 4 decimales y un intervalo de muestreo de 0.001[s]. Se ocupó el canal “TP_ADC_CH5” debido a que, anteriormente en la construcción de la tarjeta Planar Langmuir Probe, se dejó este canal en circuito abierto, para ser el “punto de prueba” de una futura calibración. Desde el osciloscopio “Tektronix MDO4054B-3” se obtuvo los valores de entrada al sistema ADC que se deseaba calibrar, además, se obtuvo el cálculo de la esperanza y la desviación estándar del voltaje impuesto. Estos 2 estimadores se escogieron por ser entregados y calculados rápidamente por el osciloscopio, además, de que entregan la información necesaria para calcular los errores más frecuentes al realizar una calibración a un ADC [18].

Se deseó caracterizar de la mejor forma esta señal de entrada al ADC y evitar los efectos ruidosos provenientes de distintas fuentes, como: la topología de construcción, los componentes utilizados, el estado electromagnético del ambiente en el momento de la medición, etc. Algunas medidas que se tomaron para evitar estos efectos fueron: utilizar conexión directa y común para todas las tomas a tierra requeridas, utilizar un brazaletes a tierra al manipular el circuito y tomar las medidas, y por último, ocupar una superficie de plano tierra debajo de todos los equipos involucrados.

Desde el monitor serial integrado en el software “Arduino IDE” se obtuvo los valores de salida al sistema ADC que se deseaba calibrar. Estos valores son las lecturas de voltaje



Figura 4.2: Foto de la fuente de alimentación que energiza la tarjeta Planar Langmuir Probe. Necesita ser alimentada con $-17[V]$, $0[V]$ y $17[V]$. Habitualmente consume $0.03[A]$ en la terminal negativa y $0.05[A]$ en la terminal positiva.

enviados por el ADC y recibidos por la tarjeta SBC. Se guardaron 22 impresiones de 1000 lecturas en un archivo de texto con extensión “.log”. Estas lecturas pueden ser interpretadas en niveles (counts), con valores posibles entre 0 y 65.535, las cuales corresponden a los valores posibles para palabras de 16 bits. Sin embargo, requieren de un tratamiento numérico para su correcta interpretación.

Se creó un código (varios scripts) en el programa de sistemas de cómputo numérico “Matlab” del desarrollador MathWorks, para leer estos archivos de datos (mediciones de entradas y salidas del ADC, además, de cálculos estadísticos como la esperanza y la varianza), generar gráficos de estas lecturas y comparar su desempeño frente a la información del fabricante.

Respecto al tratamiento que se realizó con los datos se puede mencionar lo siguiente: los datos de entrada al sistema ADC no recibieron cambios, ya que se encuentran en valores de voltaje; los datos de salida del sistema ADC son recibidos como dos palabras en niveles “counts” (son nombrados “highbyte” y “lowbyte”, de 8 bits cada uno), se concatenaron, se convirtieron en “complemento 2” con valores positivos, luego se convirtieron en “complemento 2” con valores con signo (valores positivos y negativos) y finalmente se convirtieron en valores de voltajes.

Los gráficos que se imprimieron en el programa “Matlab” fueron los siguientes:

- La “regresión lineal” de los datos de salida del ADC, respecto a los datos de entrada del ADC.
- El error entre los datos de entrada del ADC y de salida del ADC, respecto a los datos de entrada del ADC.

- El error entre los datos de salida del ADC y su regresión lineal estándar, respecto a los datos de entrada del ADC.
- La desviación estándar de los datos de entrada del ADC, respecto a los datos de entrada del ADC.
- La desviación estándar de los datos de salida del ADC, respecto a los datos de entrada del ADC.

En la Figura 4.3 se puede ver un ejemplo de como se realizaron estas representaciones. En su mayoría se graficó los valores obtenidos al procesar de alguna u otra forma los datos de salida del sistema ADC respecto a los valores de entrada del sistema ADC. Se ajustó los ejes para dejar centrado el rango de operación configurado en el sistema ADC. Todo esto con el fin de obtener valores empíricos aproximados de los errores típicos que especifican los fabricantes para caracterizar a los ADC y así poder evaluar su funcionamiento.

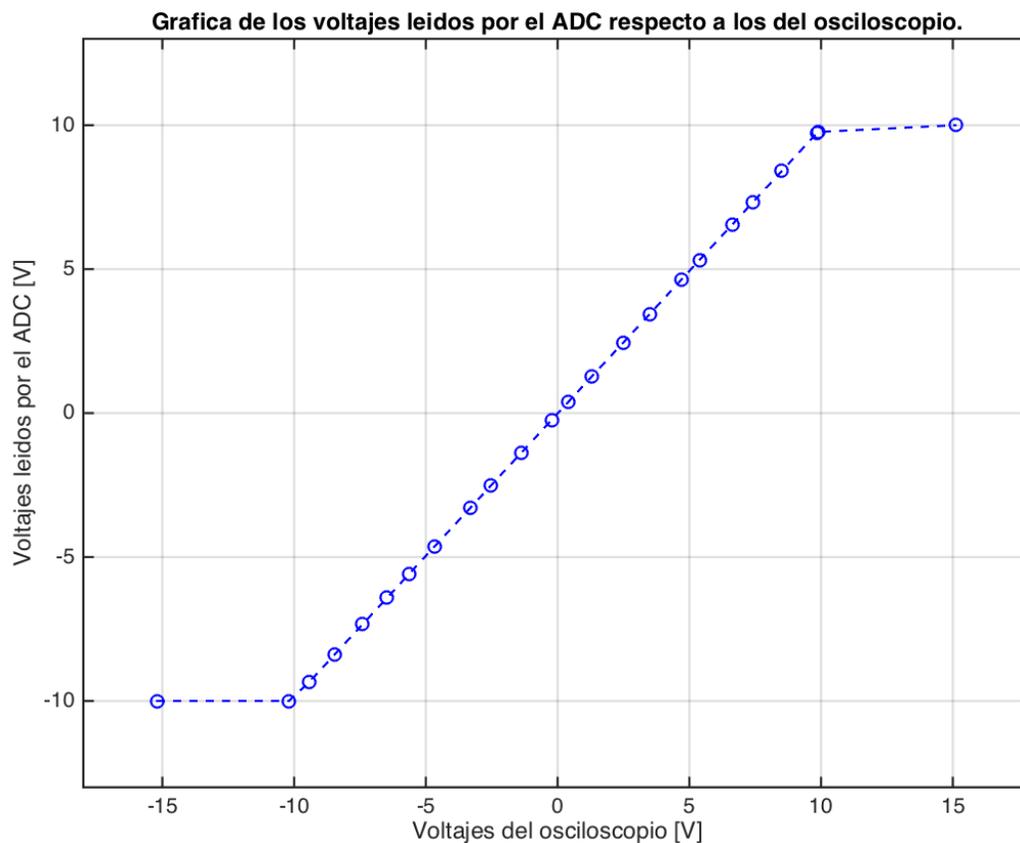


Figura 4.3: *Ejemplo de una de las gráficas que se logró con los datos obtenidos de la calibración del sistema ADC ocupando el microprocesador Arduino Due. Se muestra los valores de salida respecto a los valores de entrada.*

4.2. El montaje de la tarjeta Planar Langmuir Probe ya construida y el AVR uc3-a3 Xplained (modelo del microprocesador: AT32UC3A3256).

Se finalizó el soldado y la adhesión de las últimas partes para la tarjeta que llevaría al SBC, el detalle del trabajo es el siguiente: se incluyó el soporte de la tarjeta microSD, se incluyó los conectores para los “header” de la tarjeta AVR uc3-a3 Xplained y se incluyó el receptor del conector molex. Esta tarjeta finalizada conecta: la placa AVR uc3-a3 Xplained, una tarjeta microSD y el conector molex del ADC de la Planar Langmuir Probe.

La Figura 4.4 nos muestra el diseño digital desarrollado por ingenieros del laboratorio SPEL de la tarjeta que conecta la tarjeta SBC AVR uc3-a3 Xplained con el resto del instrumento Langmuir. Se pueden ver los caminos de cobre que conectan los distintos pines de la tarjeta SBC con el conector molex de la tarjeta Langmuir. También se puede ver los distintos planos de la tarjeta gracias al color de los caminos de cobre (azul y rojo). Se instaló el entorno para manejar el “Suchai Flight Software” (SFS) [6]. Éste corresponde al Sistema operativo GNU/Linux del tipo Unix, en específico, Ubuntu 16.04 LTS (versiones superiores no tienen soporte por el laboratorio SPEL). Se trabajó en una copia local del Suchai Flight Software, sistema operativo que fue diseñado para controlar un nanosatélite, en específico el OBC o las SBC de los payload. El repositorio y soporte viene dado a través de la plataforma de desarrollo colaborativo “GitHub”. También se debe mencionar que éste entorno fue instalado dentro del programa “VirtualBox” (software que permite correr sistemas operativos dentro de otro sistema operativo) y se ocupó la herramienta de “clonación” proporcionada por el software VirtualBox para “versionar” el código del “SFS” cada vez que se alcanzó un objetivo.

Se compiló una versión modificada del SFS y se programó (grabó) en la tarjeta AVR uc3-a3 Xplained. El compilador que se utilizó fue “GCC” (GNU Compiler Collection). Para poder programar en la tarjeta AVR uc3-a3 Xplained es necesario configurarla en modo “DFU” (Device Firmware Update), el cual se consigue manteniendo pulsado el botón integrado “SW0” y energizando la placa a través de su “USB” (Universal Serial Bus) a una fuente de alimentación externa.

Se conectó el puerto “USB” del computador a una placa “FTDI” (Future Technology Devices International) y ésta, a su vez, a la conexión “USART” (Universal Synchronous And Asynchronous Receiver-transmitter) de la tarjeta AVR uc3-a3 Xplained. Esto se realizó para leer los datos de salida del ADC que son digitalizados por la tarjeta AVR uc3-a3 Xplained. La conexión “USART” entre la placa AVR uc3-a3 Xplained y la placa “FTDI” se realizó por 3 conexiones respectivamente: desde “USART1 TXD” a “RX1”, desde “USART1 RXD” a “TX0” y desde “GND” a “GND”. Los nombres de los “pines” de la tarjeta AVR uc3-a3 Xplained se encuentran en su documentación [11].

Se implementó la impresión de las lecturas del Suchai Flight Software a través del monitor serial “CuteCom” (Graphical Serial Terminal). Se recibió estas impresiones por un puerto “USB”. En un comienzo se ocupó la velocidad de transmisión “baudrate” del monitor serial a 115.200. Esto porque las configuraciones que se ocuparon para el SFS lo ameritaban. Se modificó el archivo “main.c” del “SFS” como primera labor de programación. Si en este lugar

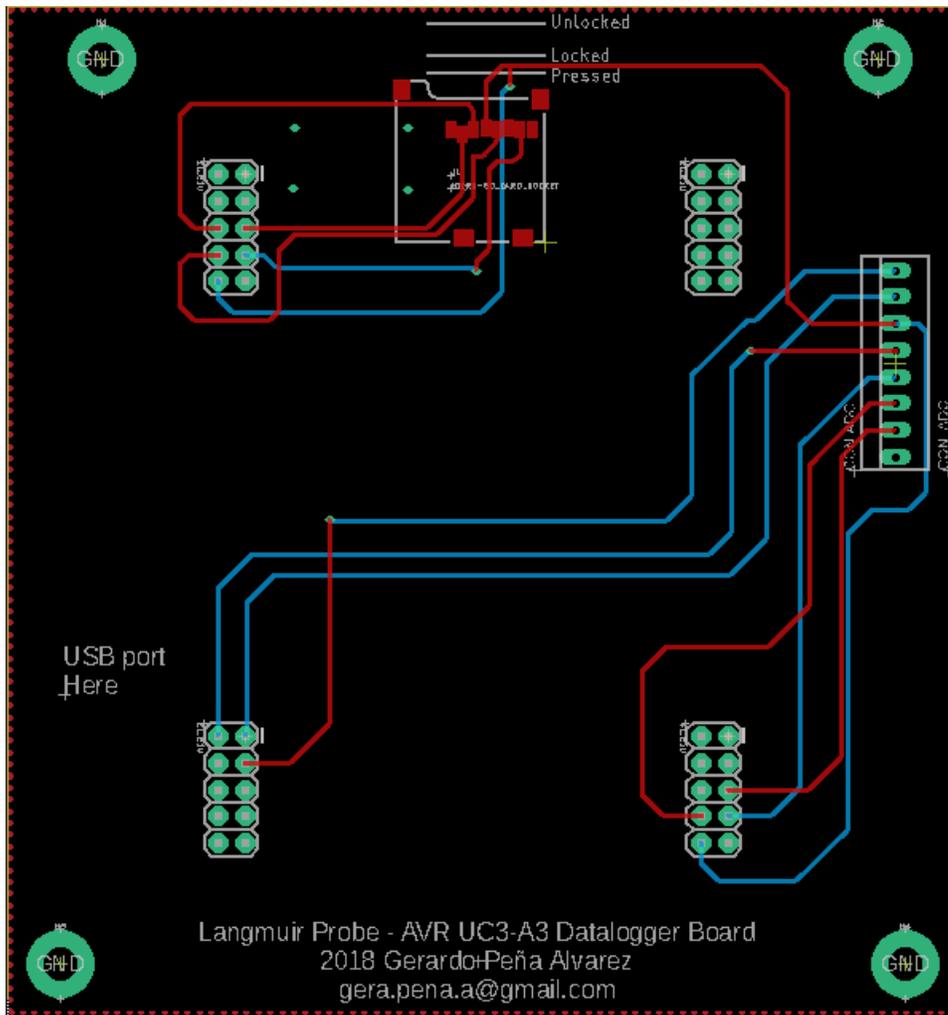


Figura 4.4: Esquema digital del diseño de la tarjeta que se terminó de montar. Esta es la conexión entre la tarjeta SBC y el resto del instrumento. El diseño y fabricación es de ingenieros del laboratorio SPEL (Gerardo Peña).

es exitosa la rutina, luego se puede trasladar todas las modificaciones a un script propio del instrumento Langmuir Probe. Esto consistió en configurar, inicializar e incluir los drivers de la comunicación “SPI”. Esto significa llamar funciones y variables declaradas y definidas en archivos “.h y .c” que pertenecen a otras carpetas del “SFS” [10].

Se logró implementar la comunicación entre la tarjeta Xplained y el ADC, para ser capaz de recibir sus lecturas, siguiendo un código de ejemplo encontrado en internet [10]. Para esto se implementó el ciclo de trabajo en el “SFS” que requiere el conversor análogo a digital para enviar sus mediciones.

En la Figura 4.5 se muestra el ciclo de trabajo del ADC, esto consiste en los estados que adquieren ciertos “pines” o terminales de interés. Algunos de estos pines deben ser configurados siguiendo este ciclo de trabajo, otros, en oposición, responden como se muestra en el ciclo a la configuración del software.

Dentro del ciclo de trabajo hay en específico una tarea llamada “BUSY”, que muestra

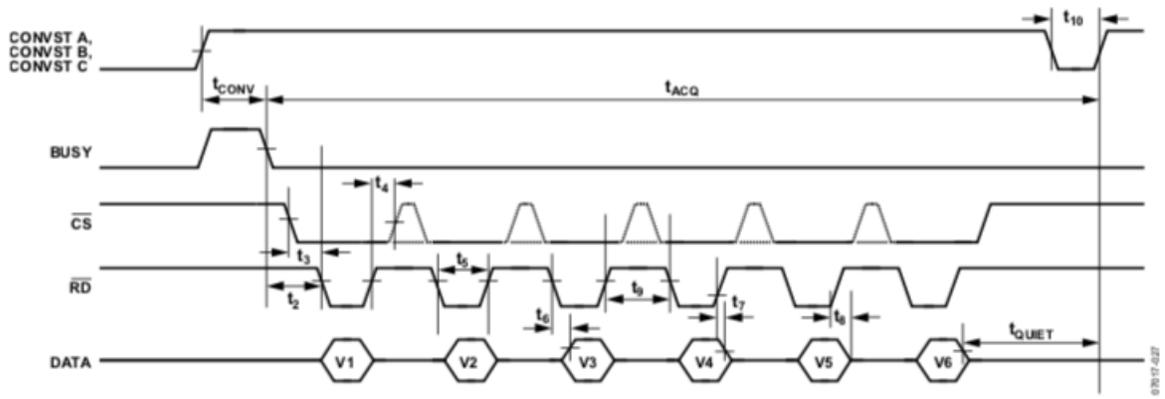


Figure 29. Parallel Interface Timing Diagram ($\overline{W}/B = 0$)

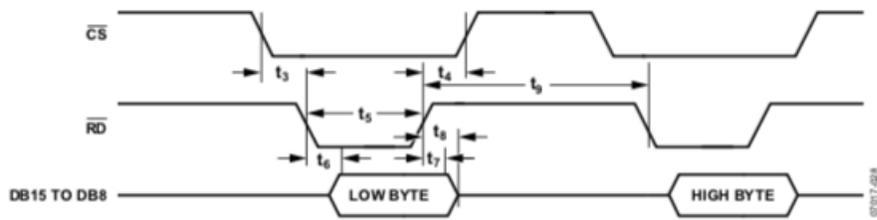


Figure 30. Parallel Interface—Read Cycle for Byte Mode of Operation ($\overline{W}/B = 1, HBEN = 0$)

Figura 4.5: La figura superior muestra el ciclo de trabajo del ADC, esto consiste en los estados que adquieren ciertos “pines” o terminales de interés. Este ciclo debe ser ejecutado para que el ADC funcione correctamente y pueda transformar las señales analógicas en lecturas digitales. Además, en la figura superior nos muestran los niveles de voltaje que adoptan los “pines” respecto al tiempo, a la izquierda se detallan los nombres de los “pines” del ADC. La figura inferior muestra el detalle de uno de los 6 mensajes correspondientes al pin “DATA”. La figura fue obtenida desde [16].

el voltaje que se podría leer en el conector “ADC_BUSY” de la tarjeta PLP. El ciclo nos muestra que este voltaje se eleva y desciende, al comienzo del ciclo de trabajo. Para obtener “la velocidad de medición del ADC” se ocupó este comportamiento y luego se calculó la frecuencia de muestreo efectiva del funcionamiento del ADC.

Se imprimió digitalmente en el ordenador a través del monitor serial “CuteCom”. Estas impresiones reflejan los datos recibidos desde el puerto USB, los cuales fueron leídos por la tarjeta AVR uc3-a3 Xplained. Estas lecturas provienen desde los seis canales de emisión/recepción del ADC.

La Figura 4.6 nos muestra la vista del monitor “CuteCom” y la impresión de los seis canales. Las impresiones van apareciendo como una concatenación de filas en sentido vertical (cada fila es una muestra de medición del ADC), dejando a las más antiguas en las posiciones mas elevada de la pantalla y a las mas actuales en la parte inferior de la pantalla. El sistema “CuteCom” también entrega la posibilidad de guardar todo lo impreso en la pantalla en un archivo de registro del tipo “.log”.

Es oportuno mencionar que se requieren dos cables USB, como se dijo más atrás. El primero va directo a la tarjeta AVR uc3-a3 Xplained y con este se graba el software en el microcontrolador, el segundo va a la tarjeta FTDI que recibe la transmisión USART para visualizar los datos en el monitor serial.

Los datos impresos se pueden pensar como tres grupos. El primer grupo corresponde a la impresión del “numero de la medición” en la que se esta mirando. El segundo grupo corresponde a las siguientes cuatro impresiones (alta ganancia, baja ganancia, temperatura 1, temperatura 2), que muestran los voltajes impuestos al ADC por los cuatro canales del lado analógico de la tarjeta Planar Langmuir Probe. El tercer grupo son las ultimas 2 impresiones que muestran los canales que se dejaron en circuito abierto. La quinta impresión (grupo tres) corresponde al quinto canal del ADC que se implementó como punto de prueba llamado “TP_ADC_CH5” según el esquemático de la placa [21].

```

140600000 255||183 0||7 42||13 42||84 0||8 203||218
140700000 0||26 0||7 42||12 42||88 0||8 203||216
140800000 0||125 0||10 42||13 42||83 0||7 203||215
140900000 255||184 0||6 42||12 42||86 0||8 203||219
141000000 0||163 0||10 42||10 42||90 0||7 203||217
141100000 255||205 0||7 42||11 42||83 0||8 203||216
141200000 0||109 0||10 42||12 42||84 0||8 203||217
141300000 255||169 0||8 42||11 42||82 0||8 203||217
141400000 0||75 0||9 42||13 42||87 0||8 203||217
141500000 0||65 0||9 42||11 42||87 0||6 203||216
141600000 0||111 0||9 42||12 42||84 0||8 203||218
141700000 0||206 0||8 42||11 42||85 0||6 203||217
141800000 255||192 0||7 42||12 42||82 0||8 203||218
141900000 0||60 0||11 42||10 42||87 0||8 203||217
142000000 0||249 0||9 42||12 42||84 0||6 203||218
142100000 255||153 0||4 42||11 42||87 0||8 203||218
142200000 255||240 0||7 42||12 42||84 0||6 203||217
142300000 255||234 0||9 42||10 42||83 0||9 203||218
142400000 0||147 0||10 42||12 42||86 0||8 203||218
142500000 0||182 0||11 42||10 42||82 0||7 203||218
142600000 255||160 0||7 42||12 42||85 0||7 203||216
142700000 0||142 0||10 42||9 42||88 0||8 203||218
142800000 0||85 0||6 42||12 42||86 0||9 203||216
142900000 0||178 0||8 42||11 42||85 0||7 203||217
143000000 255||184 0||8 42||12 42||84 0||7 203||218

```

Figura 4.6: La figura nos muestra la impresión que se realizó en el monitor serial CuteCom de los seis canales de datos leídos por la tarjeta AVR uc3-a3 Xplained desde el ADC.

Debido a que el voltaje configurado en los pines DB10, DB11 y DB12 (del ADC) fue 0[V], el rango de entrada de voltaje analógico se limita a estar entre -10[V] y 10[V]. Se deseaba trabajar en este rango de voltaje, ya que el ADC es capaz de leer en este rango en cada uno de sus canales, esto debido a la configuración ajustada en el ADC. Dentro de este rango valores el ADC presenta un comportamiento lineal. Fuera de estos valores el ADC presenta un comportamiento de saturación.

Se construyó un nuevo circuito capaz de entregar voltajes constantes en el rango de -10[V] a 10[V]. Se logró disminuir el ruido de su señal de salida ocupando filtros pasa bajos (RC). Se implementó y construyó un “Amplificador Seguidor de Voltaje” para entregar el mínimo de corriente.

La Figura 4.7 nos muestra el diseño del circuito que se construyó, ocupando un potenciómetro para variar el voltaje, filtros para el ruido en las entradas y la salida y un OPAM para el “Amplificador Seguidor de Voltaje”. Entregaba valores entre $-15[V]$ a $15[V]$. Este circuito se implementó finalmente con éxito en una protoboard y su fuente de alimentación fue desde 3 puntos de la tarjeta Planar Langmuir Probe que entregan voltajes fijos y con bajo ruido. El nombre de estos puntos son “TP_V+”, “TP_V-”, y “TP_GND” de acuerdo con su diseño [21] y entregan los siguientes voltajes en el mismo orden: $+15[V]$, $-15[V]$ y $0[V]$. El Planar Langmuir Probe se energizó utilizando una fuente de alimentación DC: GP-4303TP. La Figura 4.2 nos muestra la fuente que se utilizó.

4.2.1. Calibración en valores de voltajes constantes: generar gráficos y comparar el desempeño visto frente a la información del fabricante.

Posteriormente, con este circuito se recorrieron 21 valores equiespaciados en el rango entre $-11[V]$ y $11[V]$. Son 22 medidas equiespaciadas ya que así se puede recorrer este rango con una separación entre valores de $\pm 1[V]$, lo cual nos entrega una discretización razonable del espacio de funcionamiento. Además, se quiso obtener mediciones en la zona de operación lineal, y otros, en la zona de operación saturada.

Se midió con la sonda análoga del osciloscopio modelo “Tektronix MDO4054B-3” cada una de estas señales, las cuales provienen de los voltajes constantes impuestos, por el circuito antes mencionado, en el canal de prueba del ADC “TP_ADC_CH5”, posicionado en la tarjeta Langmuir.

Se ocupó un osciloscopio, debido a que había que tener exactitud en los valores de entrada al sistema ADC. Cada una de las mediciones del osciloscopio guardadas corresponden a una colección de datos en una ventana de tiempo. Estos datos tienen una precisión mayor o igual a 4 decimales y un intervalo de muestreo de $0.001[s]$. Se ocupó el canal “TP_ADC_CH5” debido a que, anteriormente en la construcción de la tarjeta Planar Langmuir Probe, se dejó este canal en circuito abierto, para ser el “punto de prueba” de una futura calibración.

Desde el osciloscopio “Tektronix MDO4054B-3” se obtuvo los valores de entrada al sistema ADC que se deseaba calibrar, además, se obtuvo el cálculo de la esperanza y la desviación estándar del voltaje impuesto. Estos 2 estimadores se escogieron por ser entregados y calculados rápidamente por el osciloscopio, además, de que entregan la información necesaria para calcular los errores más frecuentes al realizar una calibración a un ADC [18].

Se deseó caracterizar de la mejor forma esta señal de entrada al ADC y evitar los efectos ruidosos provenientes de distintas fuentes, como: la topología de construcción, los componentes utilizados, el estado electromagnético del ambiente en el momento de la medición, etc. Algunas medidas que se tomaron para evitar estos efectos fueron: utilizar conexión directa y común para todas las tomas a tierra requeridas, utilizar un brazaletes a tierra al manipular el circuito y tomar las medidas y, por último, ocupar una superficie de plano tierra debajo de todos los equipos involucrados.

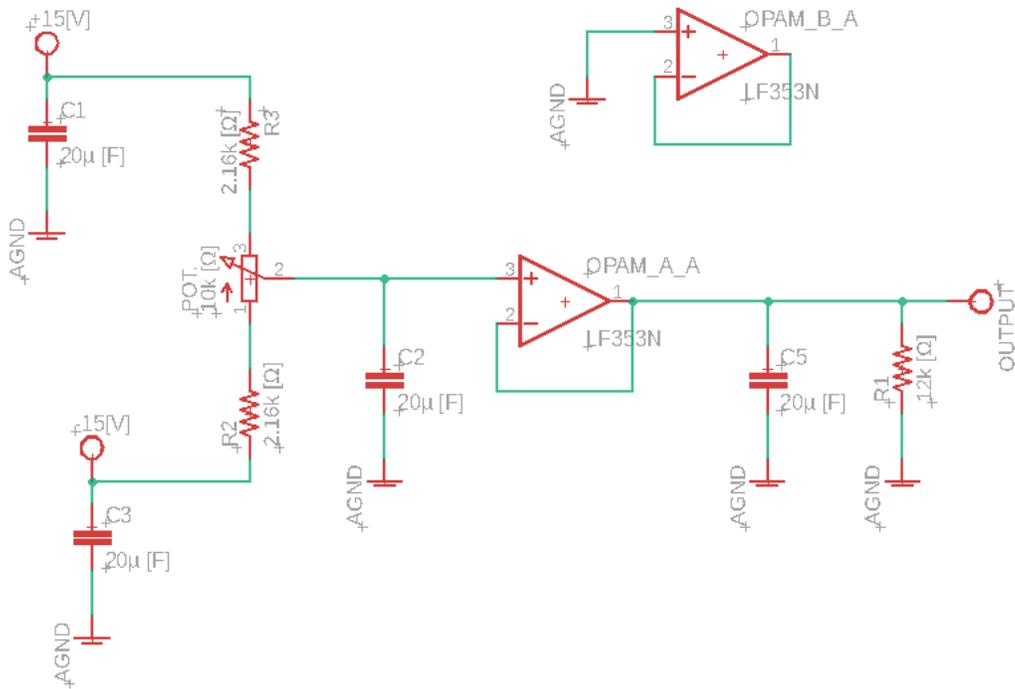


Figura 4.7: La figura nos muestra el diseño del circuito que se construyó, para entregar valores entre $-15[V]$ a $15[V]$. Se ocupó un potenciómetro, filtros y un OPAM para el “Amplificador Seguidor de Voltaje”.

A partir de las impresiones por el monitor serial “CuteCom” se obtuvo las lecturas de los valores de salida del sistema ADC, que se deseaba calibrar en valores de voltaje constante. Se guardaron 22 impresiones de 1000 lecturas cada una, en archivos con extensión “.log”. Estas lecturas pueden ser interpretadas en niveles (counts), con valores posibles entre 0 y 65.535, las cuales corresponden a los valores posibles para palabras de 16 bits. Sin embargo, requieren de un tratamiento numérico para su correcta interpretación.

Se creó un código (varios scripts) en el programa de sistemas de cómputo numérico “Matlab” del desarrollador MathWorks, para leer estos archivos de datos (mediciones de entradas y salidas del ADC, además, de cálculos estadísticos como la esperanza y la varianza), generar gráficos de estas lecturas y comparar su desempeño frente a la información del fabricante.

Respecto al tratamiento que se realizó con los datos se puede mencionar lo siguiente: los datos de entrada al sistema ADC no recibieron cambios, ya que se encuentran en valores de voltaje; los datos de salida del sistema ADC son recibidos como dos palabras en niveles “counts” (son nombrados “highbyte” y “lowbyte”, de 8 bits cada una), se concatenaron, se convirtieron en “complemento 2” con valores positivos, luego se convirtieron en “complemento 2” con valores con signo (valores positivos y negativos) y finalmente se convirtieron en valores de voltajes.

Los gráficos que se imprimieron en el programa “Matlab” fueron los siguientes:

- La “regresión lineal” de los datos de salida del ADC, respecto a los datos de entrada

del ADC.

- El error entre los datos de entrada del ADC y de salida del ADC, respecto a los datos de entrada del ADC.
- El error entre los datos de salida del ADC y su regresión lineal estándar, respecto a los datos de entrada del ADC.
- La desviación estándar de los datos de entrada del ADC, respecto a los datos de entrada del ADC.
- La desviación estándar de los datos de salida del ADC, respecto a los datos de entrada del ADC.

En la Figura 4.3 se puede ver un ejemplo de como se realizaron estas gráficas.

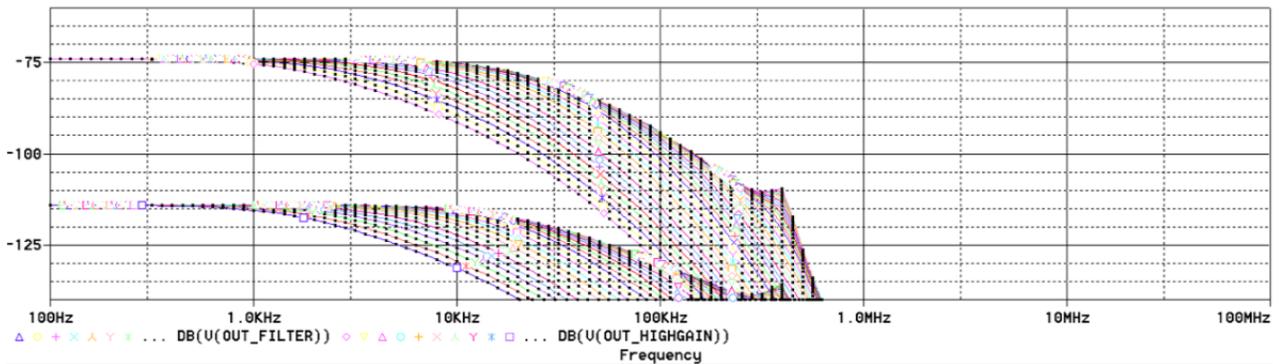


Figura 4.8: En la figura se muestra la grafica de la respuesta en frecuencia que alcanzan dos señales de salida de la parte analógica del Planar Langmuir Probe: “Low Gain” y “High Gain”. Estas señales representan las mediciones de voltaje del electrodo [20].

4.2.2. Primera calibración en frecuencia: organización del código y disminución de la tasa de muestreo de la conexión SPI.

Se logró depurar el código con el objetivo de alcanzar la frecuencia de muestreo más alta posible en el “loop principal” del software, para obtener las medidas provenientes del ADC. Para esto se eliminaron los “delays” ocupados en las rutinas de medición para la conexión “SPI”.

En un principio el código hacia las mediciones e inmediatamente después las imprimía por el monitor serial. Luego se modificó el orden de estas acciones para optimizar los tiempos. Se dispuso el código para que desarrollara las siguientes instrucciones. Primero, el código debe leer una medición del ADC. Luego, debe guardar la medición en el primer espacio de memoria de un arreglo de largo 1299 (el máximo permitido por la tarjeta del microcontrolador). Luego, debe repetir las dos instrucciones anteriores, hasta guardar una medición en el último espacio de memoria del arreglo (pensando que estuviera lleno). En ese instante el código debe imprimir el vector del arreglo por el monitor serial “CuteCom”. Finalmente se debe volver al primer espacio de memoria del arreglo (pensando que estuviera vacío) y comenzar nuevamente las instrucciones.

Se introdujo distintas señales para ver el comportamiento del ADC. Las medidas de estas señales obtenidas por el osciloscopio “Tektronix MDO4054B-3”, serían los valores de entrada al sistema ADC que se deseaba calibrar ahora en el espectro de frecuencia. Una de las señales que se introdujo fue “ruido blanco”. Esta señal fue filtrada con un filtro “pasa bajo” del tipo “RC” el cual poseía una frecuencia de corte de 17 k[Hz]. Esto con el propósito de no incluir frecuencias más altas, las cuales podrían generar efectos de “aliasing” en la banda de interés [22].

Se introdujo, además, varias señales sinusoidales con sus componentes fundamentales en: 1[Hz], 10[Hz], 30[Hz], 40[Hz], 80[Hz], 200[Hz], 500[Hz], 1k[Hz], 3k[Hz], 5k[Hz], 7k[Hz] y 10k[Hz]. Para producirlas se ocupó un generador de ondas modelo “RIGOL DG1022 Dual-Channel”.

De acuerdo con la información teórica que se tiene, las señales de salida que entrega la parte analógica del Planar Langmuir Probe [20] poseen componentes en frecuencias con una frecuencia de “Cut Off” cercana al orden de los 10^3 [Hz]. Además, se sabe que sobre esta frecuencia todas las componentes comienzan a disminuir su potencia rápidamente. La Figura 4.8 nos muestra las graficas del espectro en frecuencia para las dos señales de salida de la parte analógica del instrumento Langmuir, Low Gain y High Gain. Para un capacitor de feedback en el Amplificador de Trans-impedancia de 10 p[F] (este fue el valor que se utilizo en el instrumento Langmuir que se construyó), las frecuencias de corte son 10.438 kHz y 10.592 k[Hz] para las salidas de Low Gain y High Gain, respectivamente. Por lo tanto, la frecuencia máxima de interés estaría en 10.6 k[Hz] aproximadamente (Frecuencia de Nyquist).

Pudimos concluir que, si se desea muestrear el total de este espectro en frecuencia, se necesita un instrumento capaz de muestrear a una frecuencia igual (o superior) a 21.2k[Hz]. Específicamente más del doble. Esto debido a los problemas que surgen si no se cumple con el teorema de muestreo de Nyquist-Shannon [22].

Además, estas señales sinusoidales fueron medidas con el osciloscopio “Tektronix MDO4054B-3”. Se imprimieron y guardaron estas medidas a través de un pendrive que se conecto directamente al osciloscopio. Se guardaron 8 impresiones de 100.000 lecturas en un archivo con extensión “.csv”. Estas lecturas tienen formato en voltajes. Estas medidas serían los valores de entrada al sistema ADC que se deseaba calibrar en frecuencia.

Se imprimió por el monitor “CuteCom” los valores leídos con el microcontrolador. Por cada señal se guardó más de 4 paquetes de impresiones, con 1299 mediciones en cada una, en archivos con extensión “.log”. Esto por cada voltaje sinusoidal impuesto (8 en total) en el canal de entrada. Estas impresiones guardadas serían los valores de salida al sistema ADC que se deseaba calibrar en frecuencia. Estas lecturas pueden ser interpretadas en niveles (counts), con valores posibles entre 0 y 65.535, las cuales corresponden a los valores posibles para palabras de 16 bits. Sin embargo, requieren de un tratamiento numérico para su correcta interpretación.

Se creó un código (varios scripts) en el programa de sistemas de cómputo numérico “Matlab” del desarrollador MathWorks, para leer estos archivos de datos (mediciones de entradas y salidas del ADC), realizarles un tratamiento numérico para cambiar su formato y aplicarles ciertas funciones convenientes para obtener información de importancia y generar

gráficos de estas lecturas.

Respecto al primer tratamiento que se realizó con los datos se puede mencionar lo siguiente. Los datos de entrada al sistema ADC no recibieron cambios, ya que se encontraban en la forma deseada. Los datos de salida del sistema ADC fueron recibidos como dos palabras en niveles “counts” (son nombrados “highbyte” y “lowbyte”, de 8 bits cada una), se concatenaron, se convirtieron en “complemento 2” en valores positivos, luego se convirtieron en “complemento 2” en valores con signo (valores positivos y negativos) y finalmente se convirtieron en valores de voltajes.

Respecto al segundo tratamiento que se realizó con los datos se puede mencionar lo siguiente. Se aplicó la función “Fast Fourier Transform” para obtener las frecuencias con mayor presencia (mayor amplitud) dentro de las señales ocupadas. Después, se aplicó la función “Power Spectral Density” para obtener la distribución de potencia que compone a las señales ocupadas, distribución en cada componente del espectro de frecuencia. Todo esto con el fin de visualizar la presencia de las frecuencias fundamentales de cada señal a la entrada y salida del sistema ADC, y visualizar “la filtración en frecuencia” (o pérdidas) que genera el sistema ADC.

Los gráficos que se imprimieron en el programa “Matlab” fueron los siguientes:

- Cálculo de la función de transferencia del sistema ADC.
- Obtención de la “Transformada rápida de Fourier (FFT)” de la señal de entrada al ADC que entrega el Osciloscopio.
- Obtención de la “Power Spectral Density Estimates Using FFT” de la señal de entrada al ADC que entrega el Osciloscopio.
- Obtención de la “Transformada rápida de Fourier (FFT)” de la señal de salida al ADC que entrega el CuteCom.
- Obtención de la “Power Spectral Density Estimates Using FFT” de la señal de salida al ADC que entrega el CuteCom.

4.2.3. Segunda calibración en frecuencia: implementación de una ‘interrupción’ y disminución de la tasa de muestreo de la conexión SPI.

Se modificó el archivo “main.c” del “SFS” para configurar, inicializar y agregar los drivers necesarios para implementar una ‘Interrupción’ del tipo “Timer/Counter”. La interrupción es un recurso de algunas arquitecturas de controladores que es capaz de bloquear el “loop principal” del software. Además, en ese instante permite realizar instrucciones programables. Nosotros tomaremos las muestras de los datos en ese instante, para así lograr una frecuencia de muestreo máxima y ajustable por el usuario.

Para programar la interrupción se necesitó llamar funciones, variables, declaradas y definidas en archivos “.h y .c” que ya pertenecen al “SFS”. Como guía de todas estas modificaciones se utilizó un código libre de ejemplo de implementación de este proceso, encontrado en inter-

net y compatible con el compilador “GCC” y con la tarjeta AVR uc3a3 Xplained [13].

Se programó una rutina que ocupaba el Timer/Counter para obtener las mediciones del ADC, además, se dispuso el código con tareas específicas para el ciclo dentro del Timer/Counter y otras tareas específicas para el “loop principal” del software, lo que a continuación se detalla.

Tareas dentro del loop de la interrupción “Timer/Counter”:

- Se evalúa el valor de la variable que ‘evita el conflicto al imprimir’.
- Si se niega la variable que evita el conflicto, se toman las mediciones del ADC.
- Se guardan las mediciones en un primer arreglo de largo 1299 en la memoria.
- Cuando se completa el primer arreglo con mediciones, se copia el primer arreglo en un segundo arreglo.
- Se cambia el valor (se confirma) de la variable que ‘evita el conflicto al imprimir’.

Tareas dentro del loop principal del software:

- Se evalúa el valor de la variable que ‘evita el conflicto al imprimir’.
- Si se confirma la variable que evita el conflicto, se imprime el segundo arreglo por el monitor serial.
- Se cambia el valor (se niega) de la variable que ‘evita el conflicto al imprimir’.

Se programó la variable que ‘evita el conflicto al imprimir’, además, se localizaron los lugares de evaluación de la variable y se localizaron los lugares de cambio del valor de la variable, con la premisa de que nunca podía suceder la siguiente situación. La situación es “Que la interrupción bloquee al “SUCHAI Flight Software” cuando se este imprimiendo por el monitor serial”.

Se cambió la velocidad de transmisión “baudrate” del monitor serial a 230400 [símbolos/s]. Esto porque las nuevas configuraciones que se ocuparon para implementar la interrupción del tipo “Timer/Counter” en el “SUCHAI Flight Software” lo ameritaban. En el caso de ocupar otros “baudrate” se imprimen solo símbolos ininteligibles.

Se disminuyó la periodicidad de la interrupción “Timer/Counter” al mínimo valor que fue permitido por el funcionamiento. Esta se midió al mirar con el osciloscopio la actividad de la rutina del conector “ADC_BUSY” de la tarjeta Langmuir. Se alcanzo este valor de forma empírica, probando distintos valores para el denominador que acompaña al reloj que configura a la interrupción.

Se introdujo al sistema ADC algunas señales sinusoidales con sus componentes fundamentales en: 125[Hz], 250[Hz], 500[Hz], 1k[Hz], 2k[Hz], 4k[Hz], 8k[Hz] y 12k[Hz]. Para cada una de estas señales se realizó lo que se denominó como un “experimento”, el cual consistía en observar el comportamiento del sistema ADC frente a una señal sinusoidal, o sea, en obtener datos de entrada y de salida para el sistema ADC. Para producir las señales se ocupó un generador de ondas modelo “RIGOL DG1022 Dual-Channel”.

Además, estas señales sinusoidales fueron medidas con el osciloscopio “Tektronix MDO4054B-

3”. Se imprimieron estas medidas y se guardaron en un pendrive que se conecto directamente al osciloscopio. Se guardaron 8 impresiones de 100.000 lecturas en un archivo con extensión “.csv”. Estas lecturas tienen formato en voltajes. Estas medidas serían los valores de entrada al sistema ADC que nuevamente se deseaba calibrar en frecuencia.

Se imprimió por el monitor “CuteCom” los valores leídos por el microcontrolador. Por cada señal se guardó más de 4 paquetes de impresiones, con 1299 mediciones en cada una, en archivos con extensión “.log”. Esto por cada voltaje sinusoidal impuesto (8 en total) en el canal de entrada. Estas impresiones guardadas serían los valores de salida al sistema ADC que nuevamente se deseaba calibrar en frecuencia. Estas lecturas pueden ser interpretadas en niveles (counts), con valores posibles entre 0 y 65.535, las cuales corresponden a los valores posibles para palabras de 16 bits. Sin embargo, requieren de un tratamiento numérico para su correcta interpretación.

Se creó un código (varios scripts) en el programa de sistemas de cómputo numérico “Matlab” del desarrollador MathWorks, para leer estos archivos de datos (mediciones de entradas y salidas del ADC), realizarles un tratamiento numérico para cambiar su formato y aplicarles ciertas funciones convenientes para obtener información de importancia y generar gráficos de estas lecturas.

Respecto al primer tratamiento que se realizó con los datos se puede mencionar lo siguiente. Los datos de entrada al sistema ADC no recibieron cambios, ya que se encontraban en la forma deseada. Los datos de salida del sistema ADC fueron recibidos como dos palabras en niveles “counts” (son nombrados “highbyte” y “lowbyte”, de 8 bits cada una), se concatenaron, se convirtieron en “complemento 2” en valores positivos, luego se convirtieron en “complemento 2” en valores con signo (valores positivos y negativos) y finalmente se convirtieron en valores de voltajes.

Respecto al segundo tratamiento que se realizó con los datos se puede mencionar lo siguiente. Se aplicó la función “Fast Fourier Transform” para obtener las frecuencias con mayor presencia (mayor amplitud) dentro de las señales ocupadas. Después, se aplicó la función “Power Spectral Density” para obtener la distribución de potencia que compone a las señales ocupadas, distribución en cada componente del espectro de frecuencia. Todo esto con el fin de visualizar la presencia de las frecuencias fundamentales de cada señal a la entrada y salida del sistema ADC, y visualizar “la filtración en frecuencia” (o pérdidas) que genera el sistema ADC.

Los gráficos que se imprimieron en el programa “Matlab” fueron los siguientes:

- Cálculo de la función de transferencia del sistema ADC.
- Obtención de la “Transformada rápida de Fourier (FFT)” de la señal de entrada al ADC que entrega el Osciloscopio.
- Obtención de la “Power Spectral Density Estimates Using FFT” de la señal de entrada al ADC que entrega el Osciloscopio.
- Obtención de la “Transformada rápida de Fourier (FFT)” de la señal de salida al ADC que entrega el CuteCom.
- Obtención de la “Power Spectral Density Estimates Using FFT” de la señal de salida

al ADC que entrega el CuteCom.

Finalmente, se trasladó todas las modificaciones hechas al archivo “main.c” y al resto del software, a un script propio del payload llamado “langmuirProbe.c” y “langmuirProbe.h”.

Capítulo 5

Resultados

En esta sección se detallarán los resultados obtenidos luego de llevar a cabo la implementación antes mencionada. Los resultados fueron los que detallare a continuación. Es oportuno mencionar que los códigos están anexados al final del documento.

5.1. El montaje de la tarjeta Planar Langmuir Probe ya construido y el SBC Arduino Due (modelo del microprocesador: AT91SAM3X8E).

Se logró implementar la comunicación SPI entre la tarjeta del microcontrolador y el ADC “AD7656”, además, de la correcta lectura de los datos de los 6 canales bipolares a una tasa de muestreo máxima de 83[Hz]. Los datos correctos se lograron obtener debido a una corrección en la lectura de los bits por el canal 5.

La fila impresa por el monitor serial, que representa una muestra de medición del ADC, manifiesta un desfase en los bits de los canales. Los bits de los datos no llegan al mismo tiempo que la separación de cada canal, por lo cual se implementó un tratamiento correctivo posterior por software para obtener las palabras de 2 bytes de largo.

La figura 5.1 nos muestra un ejemplo del desfase en los bits de la impresión por el monitor serial de los canales cuatro y cinco. Se puede ver (conclusión a la que se llego analizando los datos) que el bit mas importante (Most Significant Bits) de la medición del dato por el canal cinco aparece impreso en el lugar del bit menos importancia (Least Significant Bits) de la medición del dato por el canal cuatro. Se programó un software específico para controlar la tarjeta del Planar Langmuir Probe. Este consiste en dos “scripts” del software Arduino IDE. El primero, llamado “LangmuirProbeLectura.ino”. Es el código principal en el cual se definen las variables, funciones internas y se detalla el loop principal del software. Además, se definen, configuran e inicializan los “pines” y se guardan los datos en una tarjeta microSD externa. El segundo, llamado ‘ControldelADC.ino’, se definen las funciones necesarias y especificas para los ciclos de trabajo del ADC.

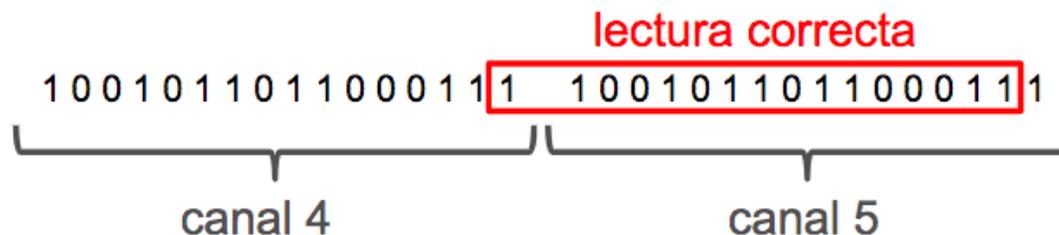


Figura 5.1: La figura es un esquema que muestra el problema (desfase) de corrimiento que presentaban las impresiones por el monitor serial.

5.1.1. Calibración en valores de voltajes constantes: generar gráficos y comparar el desempeño frente al fabricante.

Se logró calibrar el ADC en valores constantes obteniendo errores de medición del orden de tener un “Error del Voltaje de Offset” igual a 0.0042 [V], “Error en la Ganancia de Voltaje” igual a 0.1[V] (en la zona cercana a los negativos), una “Desviación Estándar para los Valores de Entrada” igual a 0.0017[V] y una “Desviación Estándar para los Valores de Salida” igual a 0.0025[V].

Es importante mencionar que, para llegar hasta esta precisión en los datos, se tuvo que realizar una corrección ocupando el valor del “Voltaje de Offset”. El “Voltaje de Offset” es un parámetro entregado en cada medición por el osciloscopio “Tektronix MDO4054B-3”, con el cual se puede obtener mayor precisión de los decimales de las mediciones.

Las figuras 5.2, 5.3, 5.4 y 5.5 son algunos de los gráficos que se lograron generar con los datos de entrada y salida al sistema. Con “sistema” nos referimos al embebido ADC “AD7656”. Estas graficas tienen el objetivo de visibilizar el comportamiento fuera de la norma e individual que pueda tener en específico esta configuración: esta construcción del instrumento Langmuir y la tarjeta SBC Arduino Due.

5.2. El montaje de la tarjeta Planar Langmuir Probe ya construido y el SBC AVR uc3-a3 Xplained (modelo del microprocesador: AT32UC3A3256).

Se logró implementar el canal de comunicación SPI0 entre la tarjeta SBC AVR uc3-a3 Xplained y el ADC “AD7656”. Además, se obtuvo la correcta lectura de los datos de los seis canales bipolares, a una tasa de muestreo máxima de 17,5k[Hz], en el caso de leer solo un canal a la vez y en el caso de leer los seis canales a la vez.

Se programó un software específico para controlar la tarjeta del Planar Langmuir Probe.

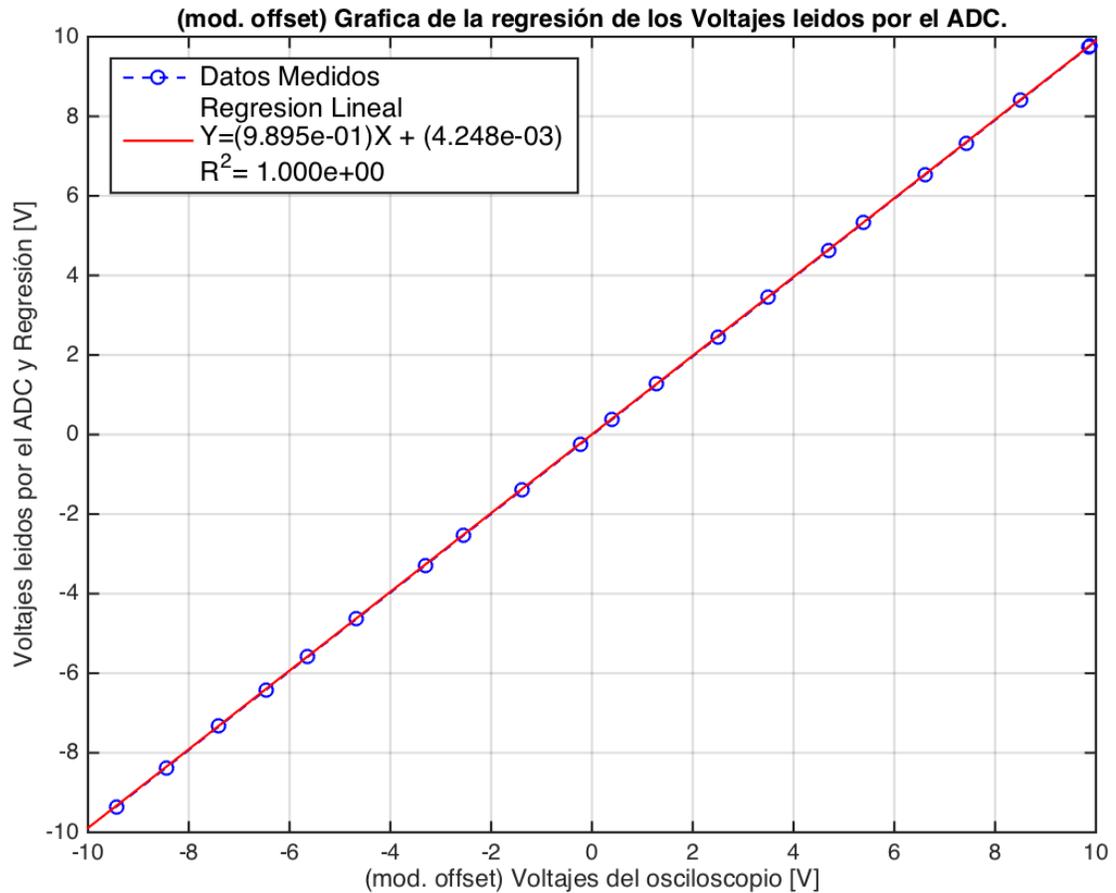


Figura 5.2: Gráfica del comportamiento de los datos de salida del ADC (leídos con la tarjeta SBC Arduino Due) y de su regresión lineal, respecto a los datos de entrada leídos desde el osciloscopio. La abreviación 'Mod. Offset' indica el aumento en la precisión de los datos gracias al 'Voltaje de Offset'. Se corrigieron los decimales de cada medida.

Este consiste en una serie de “scripts” modificados desde el “Suchai Flight Software”[6]. Se incluyeron archivos “.c” necesarios para el funcionamiento en la carpeta que contiene al 'main.c' y se incluyeron archivos “.h” necesarios para el funcionamiento en la carpeta 'Include' de esta misma locación.

Se crearon dos archivos propios del payload: el primero, llamado “langmuirProbe.c”, en donde se definen las funciones necesarias, se inicializan y se define el bucle de trabajo principal del payload; el segundo, llamado “langmuirProbe.h”, en donde se citan las librerías necesarias, se definen y configuran los pines a utilizar de la placa y se declaran las funciones propias. Se modificó el script 'main.c': se incluyó el llamado al archivo “langmuirProbe.h”. También se llamó a dos funciones definidas en el archivo “langmuirProbe.c”: “configLP()” la cual hace las configuraciones necesarias previa para comunicarse con el ADC y “measureLP()” la cual realiza las lecturas de las mediciones del ADC.

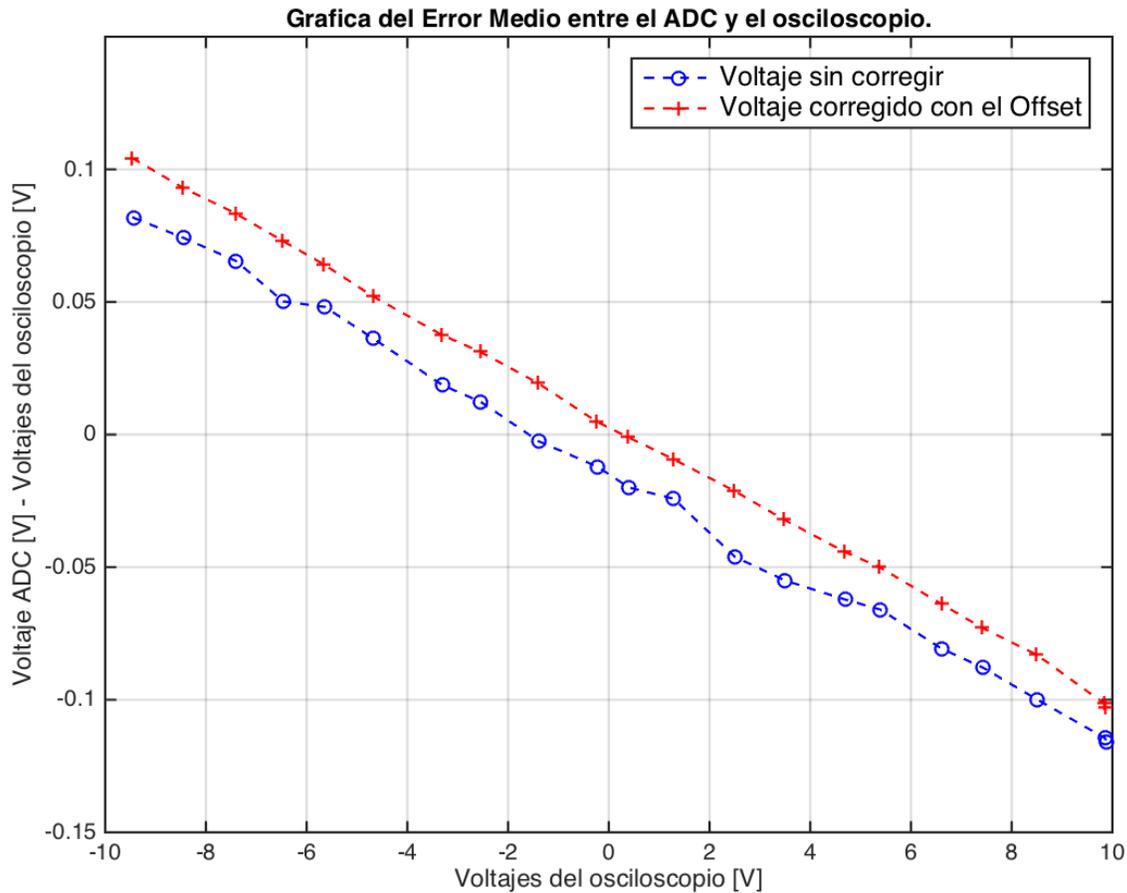


Figura 5.3: Gráfica del Error Medio entre los datos de salida del ADC (leídos con la tarjeta SBC Arduino Due) y los datos de entrada al ADC (leídos por el osciloscopio), respecto a los datos de entrada al ADC. Se muestran dos curvas: la curva sin corregir y la curva corregida con los Voltajes Offset entregados por el osciloscopio.

5.2.1. Calibración en valores de voltajes constantes: generar gráficos y comparar el desempeño frente al fabricante.

Se logró calibrar el ADC en valores constantes obteniendo errores de medición del orden de tener un “Error del Voltaje de Offset” igual a 0.0036 [V], “Error en la Ganancia de Voltaje” igual a 0.1[V] (en la zona cercana a los negativos), una “Desviación Estándar para los Valores de Entrada” igual a 0.003[V] y una “Desviación Estándar para los Valores de Salida” igual a 0.005[V].

Es importante mencionar que, para llegar hasta esta precisión en los datos, se tuvo que realizar una corrección ocupando el valor del “Voltaje de Offset”. El “Voltaje de Offset” es un parámetro entregado en cada medición por el osciloscopio “Tektronix MDO4054B-3”, con el cual se puede obtener mayor precisión de los decimales de las mediciones.

Las figuras 5.6, 5.7, 5.8, 5.9 y 5.10 son algunos de los gráficos que se lograron gene-

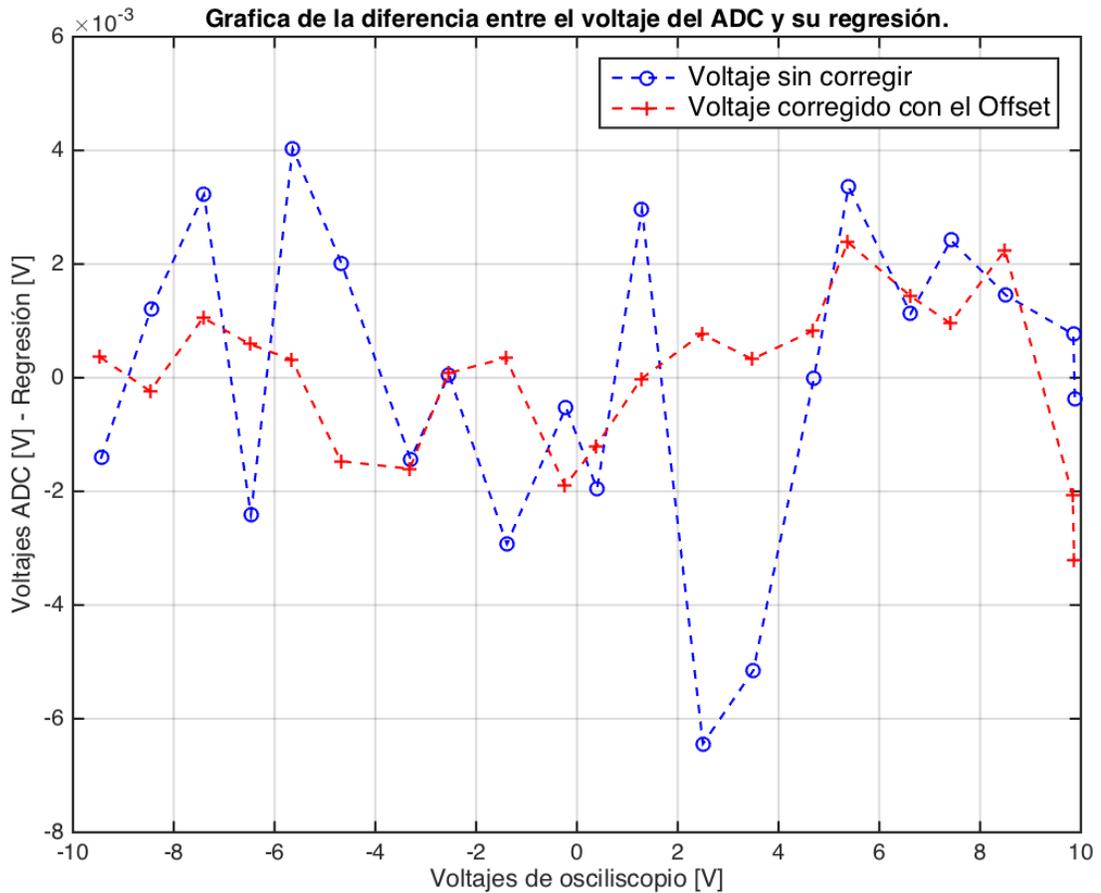


Figura 5.4: Gráfica de la diferencia entre los datos de salida del ADC (leídos con la tarjeta SBC Arduino Due) y su regresión lineal, respecto a los datos de entrada al ADC (leídos por el osciloscopio). Se muestran dos curvas: la curva sin corregir y la curva corregida con los Voltajes Offset entregados por el osciloscopio.

rar con los datos de entrada y salida al sistema. Con “sistema” nos referimos al embebido ADC “AD7656”. Estas graficas tienen el objetivo de visibilizar el comportamiento fuera de la norma e individual que pueda tener en específico esta configuración: esta construcción del instrumento Langmuir y la tarjeta AVR uc3-a3 Xplained.

5.2.2. Primera calibración en frecuencia: organización del código y disminución de la tasa de sampleo de la conexión SPI.

Se logró depurar el código en el “loop principal” del software SFS, para obtener las medidas provenientes del ADC. Para esto se eliminaron los “delays” ocupados en las rutinas de medición para la conexión “SPI”. Se dispuso el código para que desarrollara las siguientes instrucciones. Primero, el código debe leer una medición del ADC. Luego, debe guardar la medición en el primer espacio de memoria de un arreglo de largo 1299 (el máximo permitido

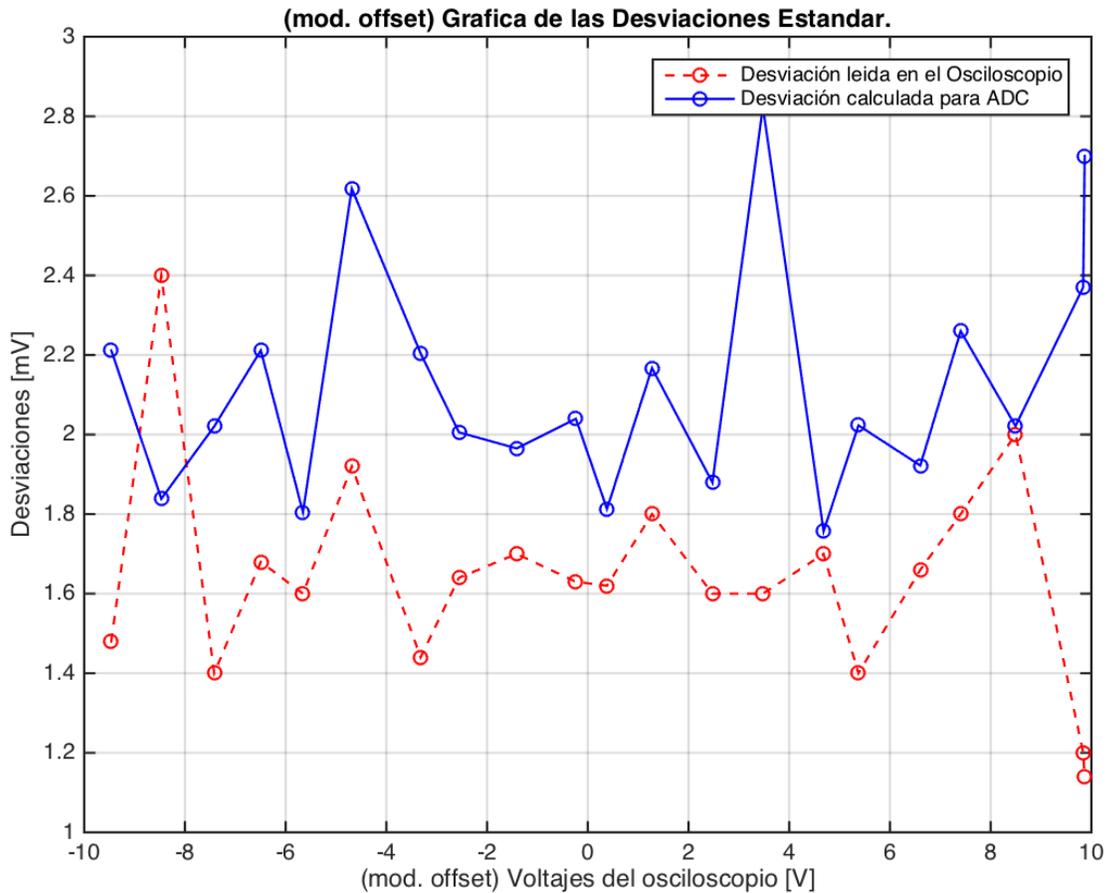


Figura 5.5: Gráfica de las desviaciones estándar de los datos de entrada al ADC (leídos por el osciloscopio) y de los datos de salida del ADC (leídos con la tarjeta SBC Arduino Due), respecto a los datos de entrada al ADC. Todos los valores son positivos y su magnitud bordea los [mV].

por la tarjeta del microcontrolador). Luego, debe repetir las dos instrucciones anteriores, hasta guardar una medición en el ultimo espacio de memoria del arreglo (pensando que esta lleno). En ese instante el código debe imprimir el vector del arreglo por el monitor serial “CuteCom”. Finalmente se debe volver al primer espacio de memoria del arreglo (pensando que este vacío) y comenzar nuevamente las instrucciones.

Con este procedimiento en el código se logró calibrar el ADC en frecuencia, dentro de un rango máximo de 8.7k[Hz] (Frecuencia de Nyquist). Frecuencias más elevadas a los 8.7k[Hz] son completamente alteradas, debido a que la máxima frecuencia de muestreo obtenida fue 17.5k[HZ].

Se introdujo distintas señales para ver el comportamiento del ADC. Una de las señales que se introdujo fue “ruido blanco”. Esta señal fue filtrada con un filtro “pasa bajo” del tipo “RC” el cual poseía una frecuencia de corte de 17 k[Hz]. Esto con el propósito de no incluir frecuencias más altas, las cuales podrían generar efectos de “aliasing” en la banda de interés

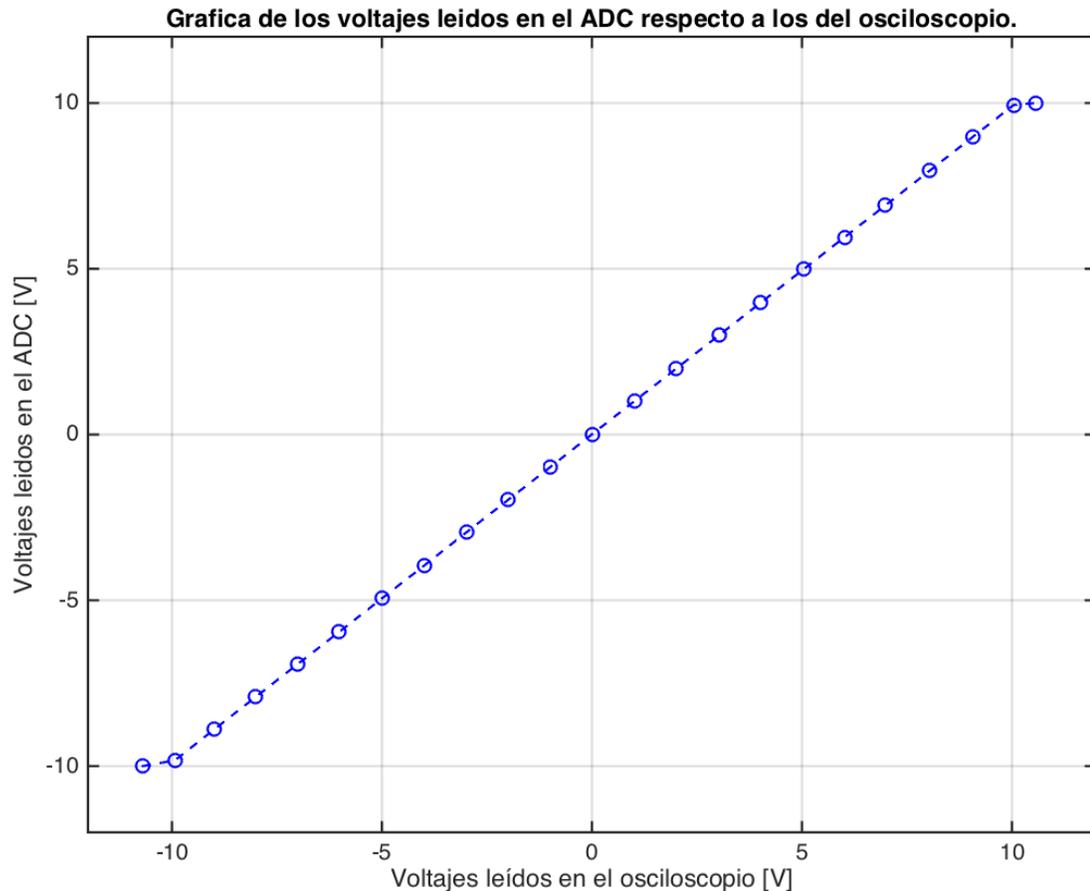


Figura 5.6: Gráfica del comportamiento de los datos de salida del ADC (leídos con la tarjeta SBC AVR uc3-a3 Xplained), respecto a los datos de entrada leídos desde el osciloscopio. Se pueden ver las dos zonas grandes de operación: la zona lineal ($|V_{out}| \leq 10$ [V]), en donde tiene un comportamiento afín ($f(x) \simeq x$) y la zona saturada, en los extremos donde se aplana la curva ($|V_{out}| \geq 10$ [V]).

[22].

Las figuras 5.11, 5.12, 5.13 y 5.14 nos muestran graficas del espectro en frecuencia de las mediciones que se realizaron para las sinusoidales con las siguientes frecuencias fundamentales: 0.08k[Hz], 0.2k[Hz], 0.5k[Hz], 1k[Hz], 3k[Hz], 5k[Hz], 7k[Hz] y 10k[Hz]. Para producirlas se ocupó un generador de ondas modelo “RIGOL DG1022 Dual-Channel”. Se tomaron estos valores por ser relativamente equiespaciados a lo largo del posible espectro de frecuencia.

Se creó un código (varios scripts) en el programa “Matlab” para leer estos archivos (mediciones de entradas y salidas del ADC), aplicarles ciertas funciones convenientes para obtener información de importancia y generar gráficos de estas lecturas.

Respecto al tratamiento que se realizó con los datos se puede mencionar lo siguiente. Se aplicó la función “Fast Fourier Transform” para obtener las frecuencias con mayor presencia (mayor amplitud) dentro de las señales ocupadas. Después, se aplicó la función “Power Spec-

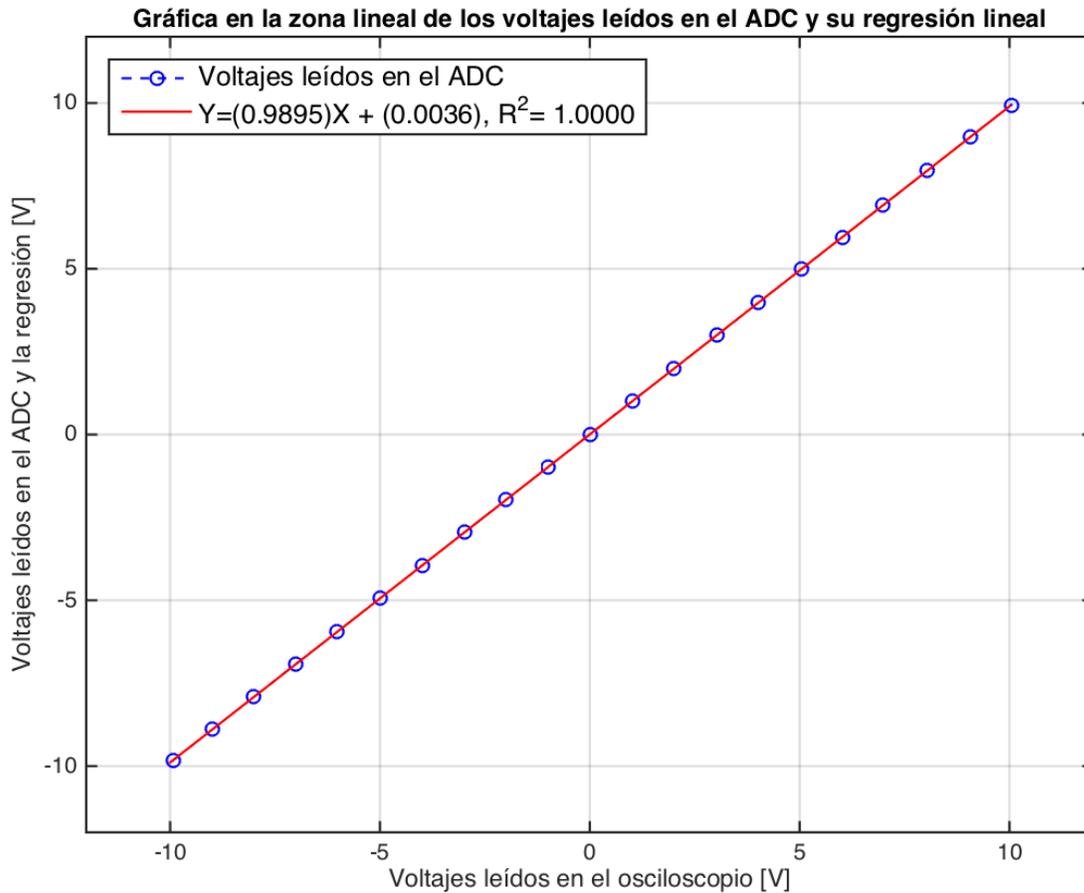


Figura 5.7: Gráfica del comportamiento de los datos de salida del ADC (leídos con la tarjeta SBC AVR uc3-a3 Xplained) y de su regresión lineal, respecto a los datos de entrada leídos desde el osciloscopio. Se corrigieron los decimales de cada medida gracias al 'Voltaje de Offset' entregados por el osciloscopio.

tral Density” para obtener la distribución de potencia que compone a las señales ocupadas, distribución en cada componente del espectro de frecuencia. Todo esto con el fin de visualizar la presencia de las frecuencias fundamentales de cada señal a la entrada y a la salida del sistema ADC. Además, se visualizó “la filtración en frecuencia” (o pérdidas) que le podría generar el sistema ADC a las señales que lo atraviesan. Por último se calculó la “Función de Transferencia” del sistema ADC y se realizó una interpolación del tipo “cubic spline” (que crea una función del tipo $f(x) = a_1 \cdot e^{-\frac{x-b_1}{c_1}} + a_2 \cdot e^{-\frac{x-b_2}{c_2}}$, con parámetros $a_1 = 219,9$, $b_1 = -7,88 \cdot 10^4$, $c_1 = 3,58 \cdot 10^4$, $a_2 = 1,783$, $b_2 = 8672$ y $c_2 = 8991$) que tiene un ajuste muy cercano a los puntos de amplitud máximas en frecuencia de las señales ocupadas (que no sufrieron alteraciones de corrimiento en frecuencia) 5.15.

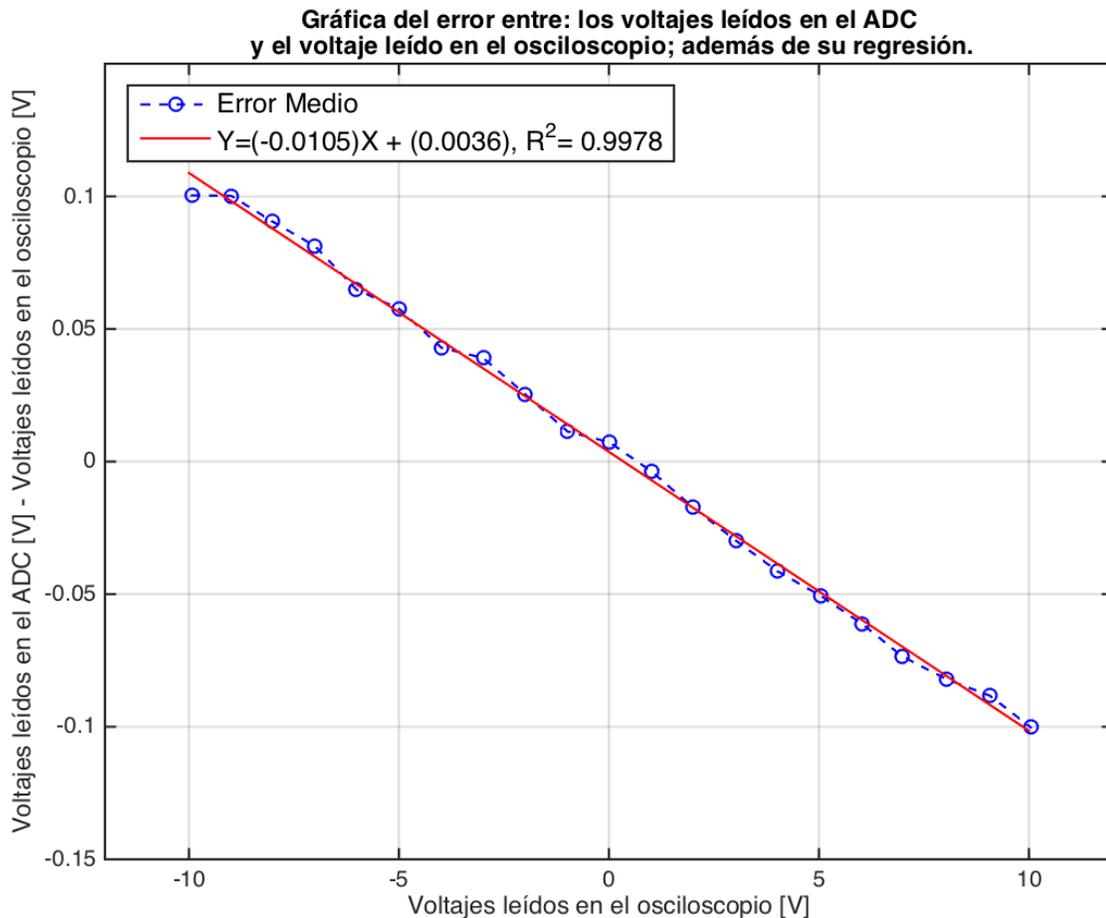


Figura 5.8: Gráfica del *Error Medio* entre los datos de salida del ADC (leídos con la tarjeta SBC AVR uc3-a3 Xplained) y los datos de entrada al ADC (leídos por el osciloscopio), respecto a los datos de entrada al ADC. Además, se graficó la regresión lineal de la curva. Se corrigieron los decimales de cada medida gracias al 'Voltaje de Offset' entregados por el osciloscopio.

5.2.3. Segunda calibración en frecuencia: implementación de una 'interrupción' y disminución de la tasa de sampleo de la conexión SPI.

Se programó un código basado en el Suchai Flight Software, compatible con el compilador "GCC" y con la tarjeta SBC AVR uc3a3 Xplained. Se modificó el archivo "main.c" del "SFS" para configurar, inicializar y agregar los drivers necesarios para implementar una 'Interrupción' del tipo "Timer/Counter" y se deseó tomar las lecturas de las mediciones del ADC justo en el instante del bloqueo al "loop principal" del software, para así obtener las mediciones con una periodicidad ajustable por el usuario desde la herramienta Timer/Counter.

Se dispuso el código para que desarrollara las siguientes instrucciones. Dentro del loop de la interrupción "Timer/Counter" se realizaron las siguientes tareas. 1) Se debe evaluar el

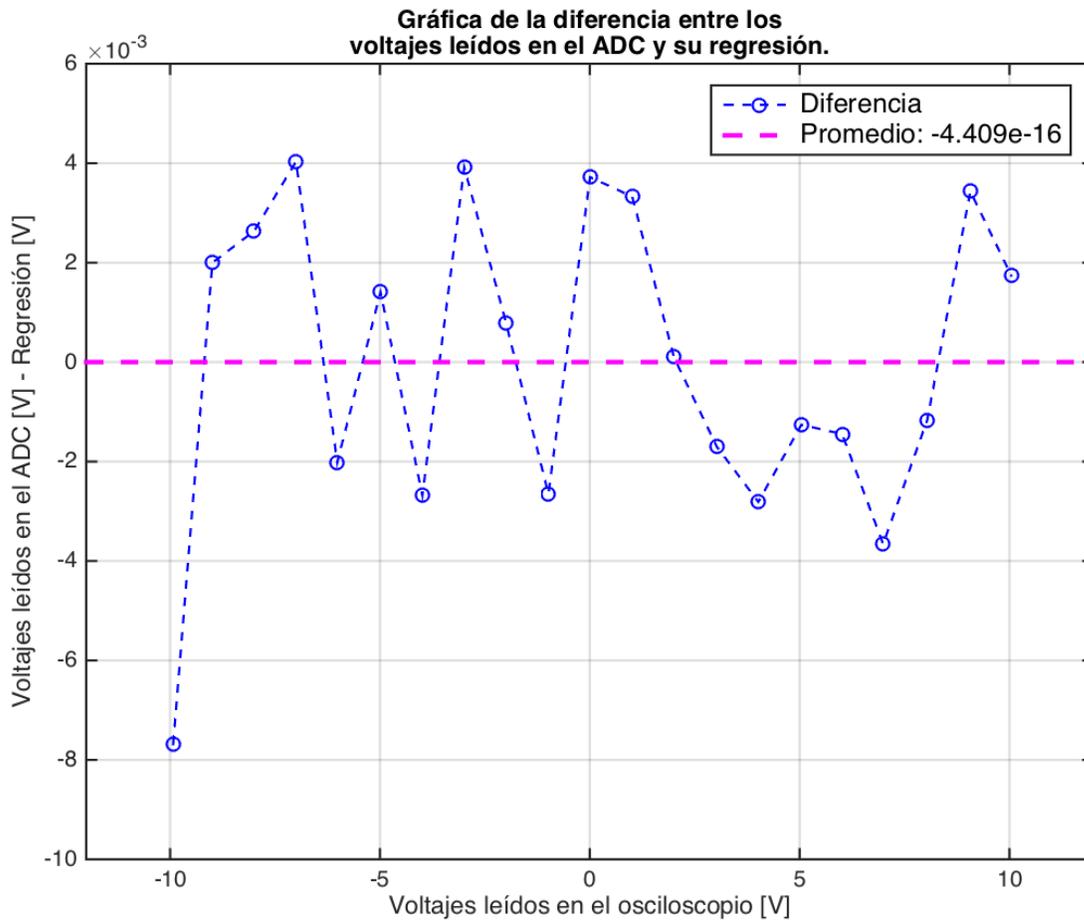


Figura 5.9: Gráfica de la diferencia entre los datos de salida del ADC (leídos con la tarjeta SBC AVR uc3-a3 Xplained) y la regresión lineal que se puede construir de los datos de salida del ADC, respecto a los datos de entrada al ADC (leídos por el osciloscopio). Además, se graficó el promedio de las mediciones. Se corrigieron los decimales de cada medida gracias al 'Voltaje de Offset' entregados por el osciloscopio.

valor de la variable que 'evita el conflicto al imprimir'. 2) Si se niega la variable que evita el conflicto, se debe tomar la medición del ADC. 3) Las mediciones se deben guardar en un primer arreglo de largo 1299 en la memoria. 4) Cuando se completa el primer arreglo con mediciones, se debe copiar el primer arreglo en un segundo arreglo. 5) Se debe cambiar el valor (se confirma) de la variable que 'evita el conflicto al imprimir'. En paralelo, dentro del "loop principal" del software se realizaron las siguientes tareas. 1) Se debe evaluar el valor de la variable que 'evita el conflicto al imprimir'. 2) Si se confirma la variable que evita el conflicto, se debe imprimir el segundo arreglo por el monitor serial. 3) Se debe cambiar el valor (se niega) de la variable que 'evita el conflicto al imprimir'.

Se programó la variable que 'evita el conflicto al imprimir', además, se localizaron los lugares de evaluación de la variable y se localizaron los lugares de cambio del valor de la variable, con la premisa de que nunca podía suceder la siguiente situación. La situación es "Que la interrupción bloquee al "SUCHAI Flight Software" cuando se este imprimiendo por

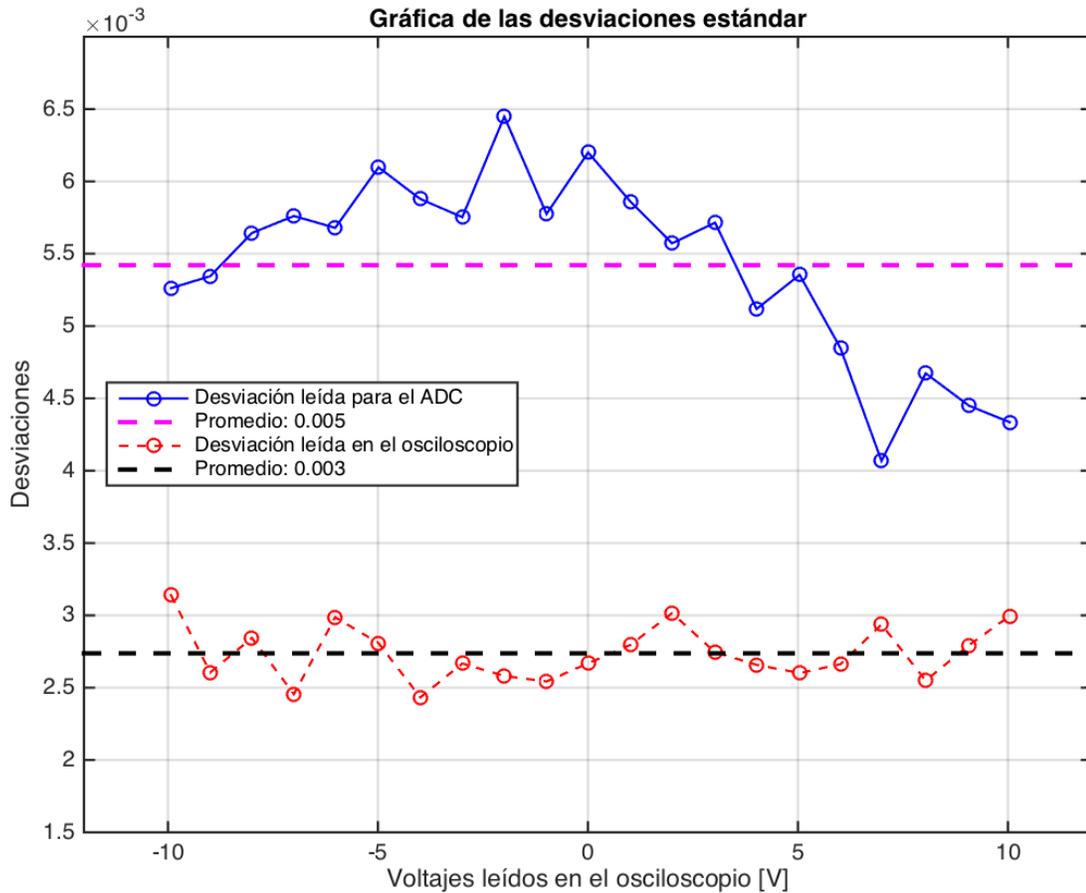


Figura 5.10: Gráfica de la curva de la desviación estándar de los datos de entrada al ADC (leídos por el osciloscopio) y grafica de la curva de la desviación estándar de los datos de salida del ADC (leídos con la tarjeta SBC AVR uc3-a3 Xplained), respecto a los datos de entrada al ADC. Todos los valores son positivos y su magnitud bordea los [mV]. Además, se graficó el promedio de cada curva de mediciones.

el monitor serial”.

Con este procedimiento en el código se logró calibrar el ADC en frecuencia, dentro de un rango máximo de 6k[Hz] (Frecuencia de Nyquist). Frecuencias más elevadas a los 6k[Hz] son completamente alteradas, debido a que la máxima frecuencia de muestreo obtenida fue 12k[Hz].

Las figuras 5.16, 5.17, 5.18 y 5.19 nos muestran graficas del espectro en frecuencia de las mediciones que se realizaron para las sinusoidales con las siguientes frecuencias fundamentales: 0.125k[Hz], 0.25k[Hz], 0.5k[Hz], 1k[Hz], 2k[Hz], 4k[Hz], 8k[Hz] y 12k[Hz]. Para producirlas se ocupó un generador de ondas modelo “RIGOL DG1022 Dual-Channel”. Se tomaron estos valores por ser relativamente equiespaciados a lo largo del posible espectro de frecuencia.

Se creó un código (varios scripts) en el programa “Matlab” para leer estos archivos (mediciones de entradas y salidas del ADC), aplicarles ciertas funciones convenientes para obtener

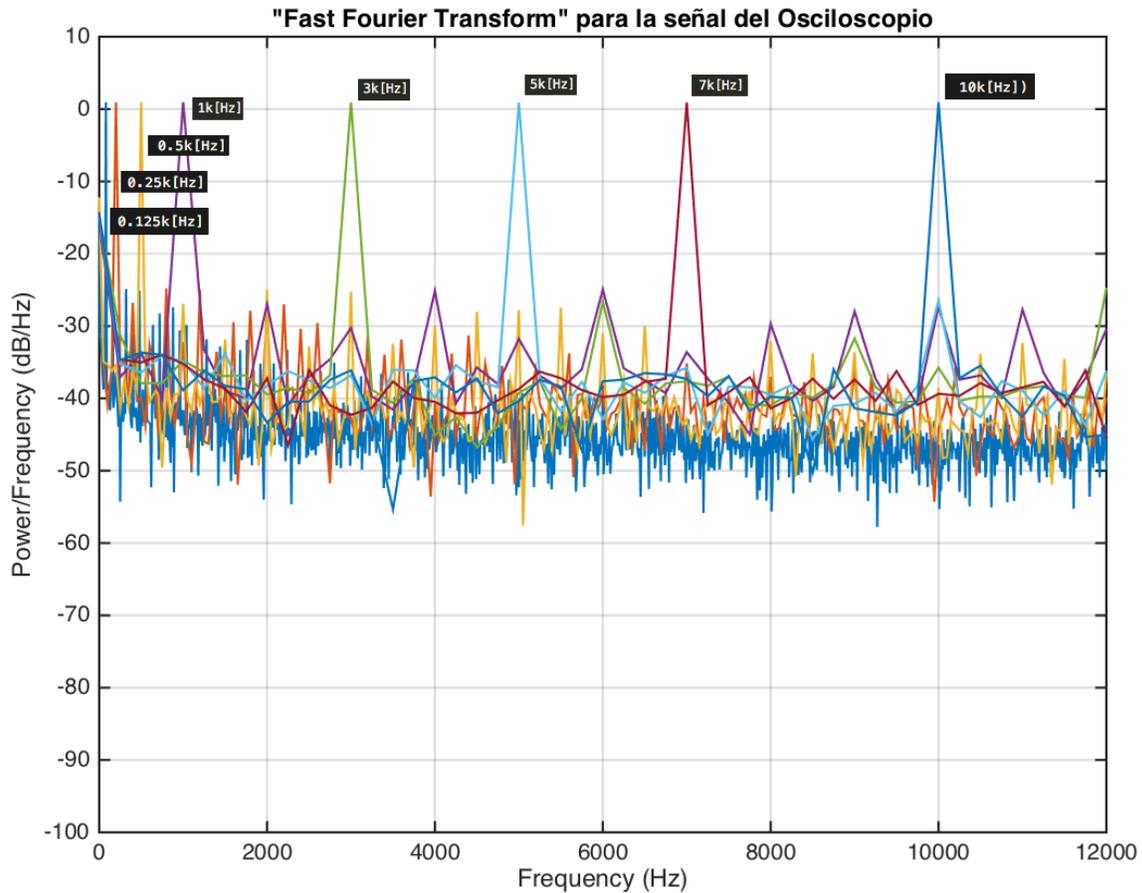


Figura 5.11: Gráfica conjunta del comportamiento de la función “Fast Fourier Transform” calculada para las mediciones de las ocho señales de entrada al ADC (leídas desde el osciloscopio), respecto al espectro de frecuencia. Cada curva tiene un color distinto y se puede ver la componente fundamental en frecuencia de esa señal (0.08k[Hz], 0.2k[Hz], 0.5k[Hz], 1k[Hz], 3k[Hz], 5k[Hz], 7k[Hz] y 10k[Hz]).

información de importancia y generar gráficos de estas lecturas.

Respecto al tratamiento que se realizó con los datos se puede mencionar lo siguiente. Se aplicó la función “Fast Fourier Transform” para obtener las frecuencias con mayor presencia (mayor amplitud) dentro de las señales ocupadas. Después, se aplicó la función “Power Spectral Density” para obtener la distribución de potencia que compone a las señales ocupadas, distribución en cada componente del espectro de frecuencia. Todo esto con el fin de visualizar la presencia de las frecuencias fundamentales de cada señal a la entrada y a la salida del sistema ADC. Además, se visualizó “la filtración en frecuencia” (o pérdidas) que le podría generar el sistema ADC a las señales que lo atraviesan. Por ultimo se calculó la “Función de Transferencia” del sistema ADC y se realizó una interpolación del tipo “cubicspline” (que crea una función del tipo $f(x) = a_1 \cdot e^{-\frac{x-b_1}{c_1}} + a_2 \cdot e^{-\frac{x-b_2}{c_2}}$, con parámetros $a_1 = 2,226$, $b_1 = 646,7$, $c_1 = 493,5$, $a_2 = 2,006$, $b_2 = 2681$ y $c_2 = 1349$) que tiene un ajuste muy cercano a los puntos de amplitud máximas en frecuencia de las señales ocupadas (que no sufrieron

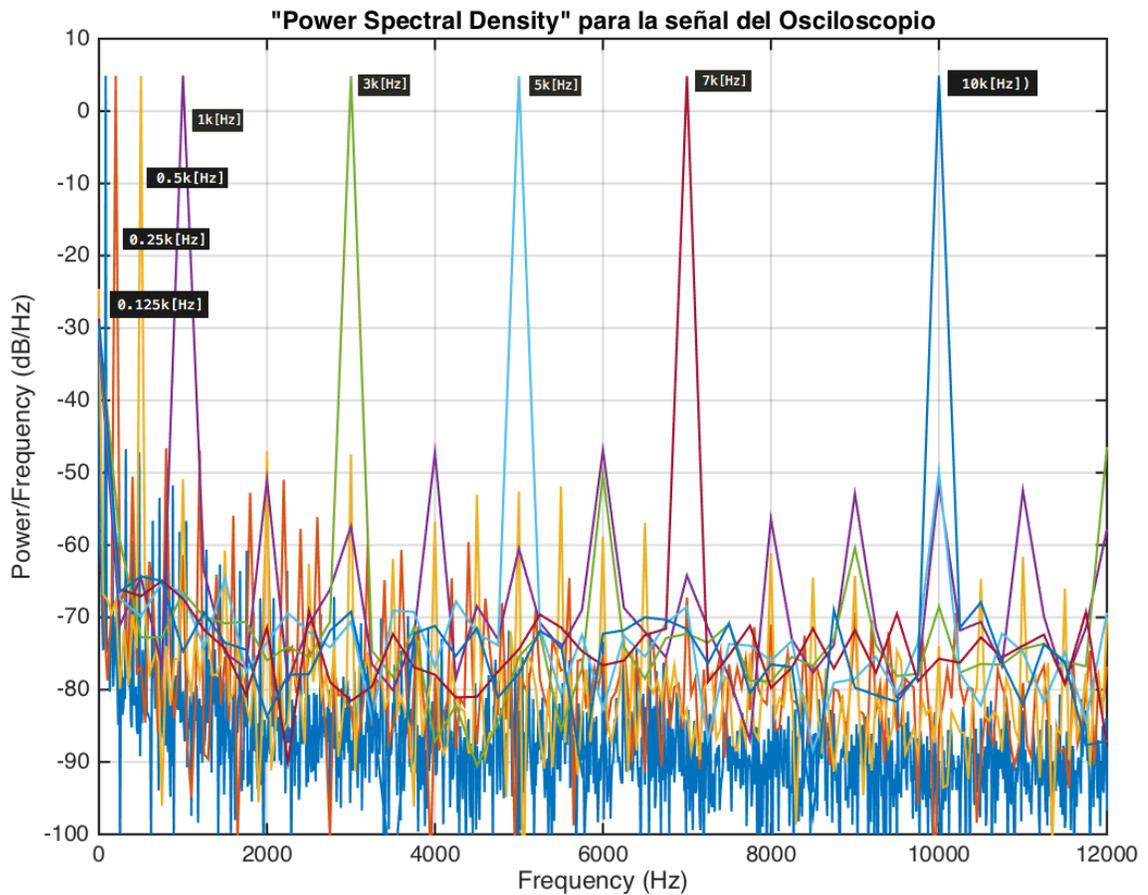


Figura 5.12: Gráfica conjunta del comportamiento de la función “Power Spectral Density” calculada para las mediciones de las ocho señales de entrada al ADC (leídas desde el osciloscopio), respecto al espectro de frecuencia. Cada curva tiene un color distinto y se puede ver la componente fundamental en frecuencia de esa señal (0.08k[Hz], 0.2k[Hz], 0.5k[Hz], 1k[Hz], 3k[Hz], 5k[Hz], 7k[Hz] y 10k[Hz]).

alteraciones de corrimiento en frecuencia) 5.20.

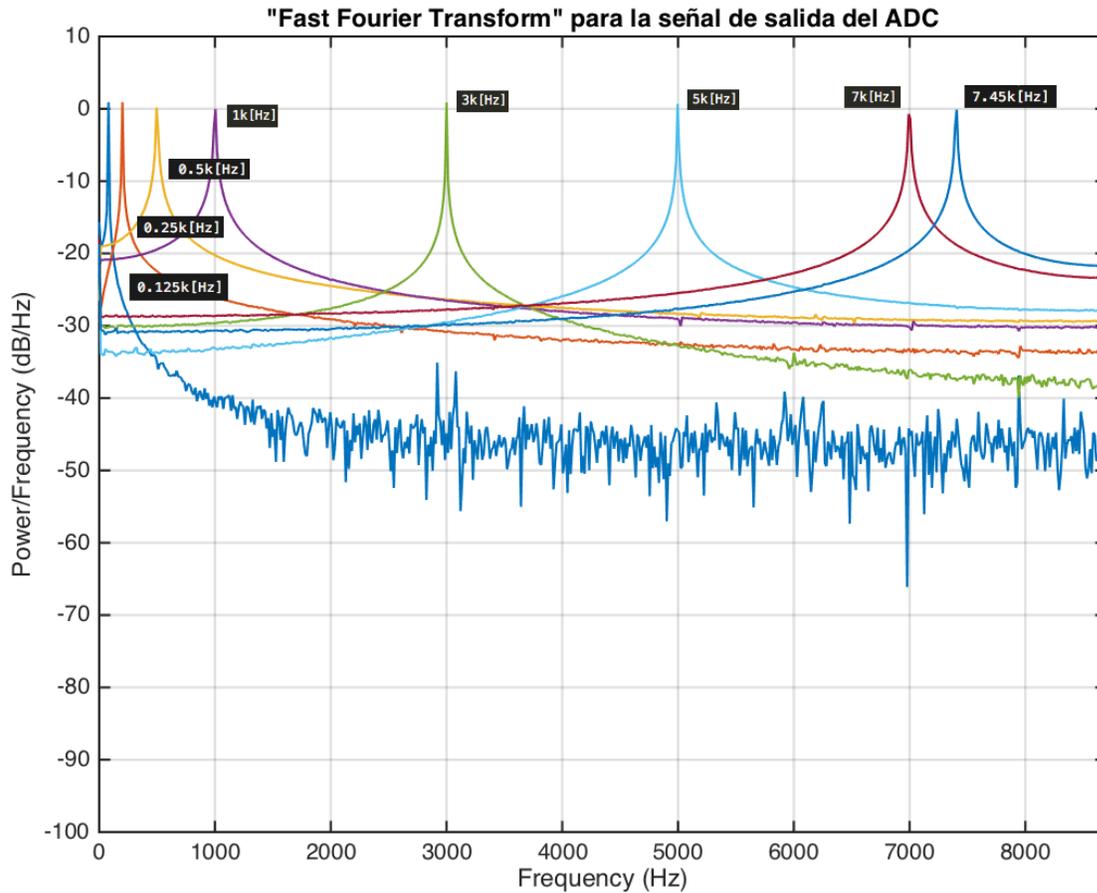


Figura 5.13: Gráfica conjunta del comportamiento de la función “Fast Fourier Transform” calculada para las mediciones de las ocho señales de salida del ADC (leídas con la tarjeta SBC AVR uc3-a3 Xplained e impresas desde el monitor serial “CuteCom”). Cada curva tiene un color distinto y se puede ver la componente fundamental en frecuencia de esa señal (0.08k[Hz], 0.2k[Hz], 0.5k[Hz], 1k[Hz], 3k[Hz], 5k[Hz] y 7k[Hz]). Además, se puede apreciar la aparición de una nueva curva con una frecuencia fundamental en 7.45k[Hz] aproximadamente. Esta curva es el resultado de la distorsión de la curva con una frecuencia fundamental en 10k[Hz] respectivamente. Esto es debido a que posee una frecuencia fundamental más altas que la frecuencia máxima de calibración (8.7k[Hz]).

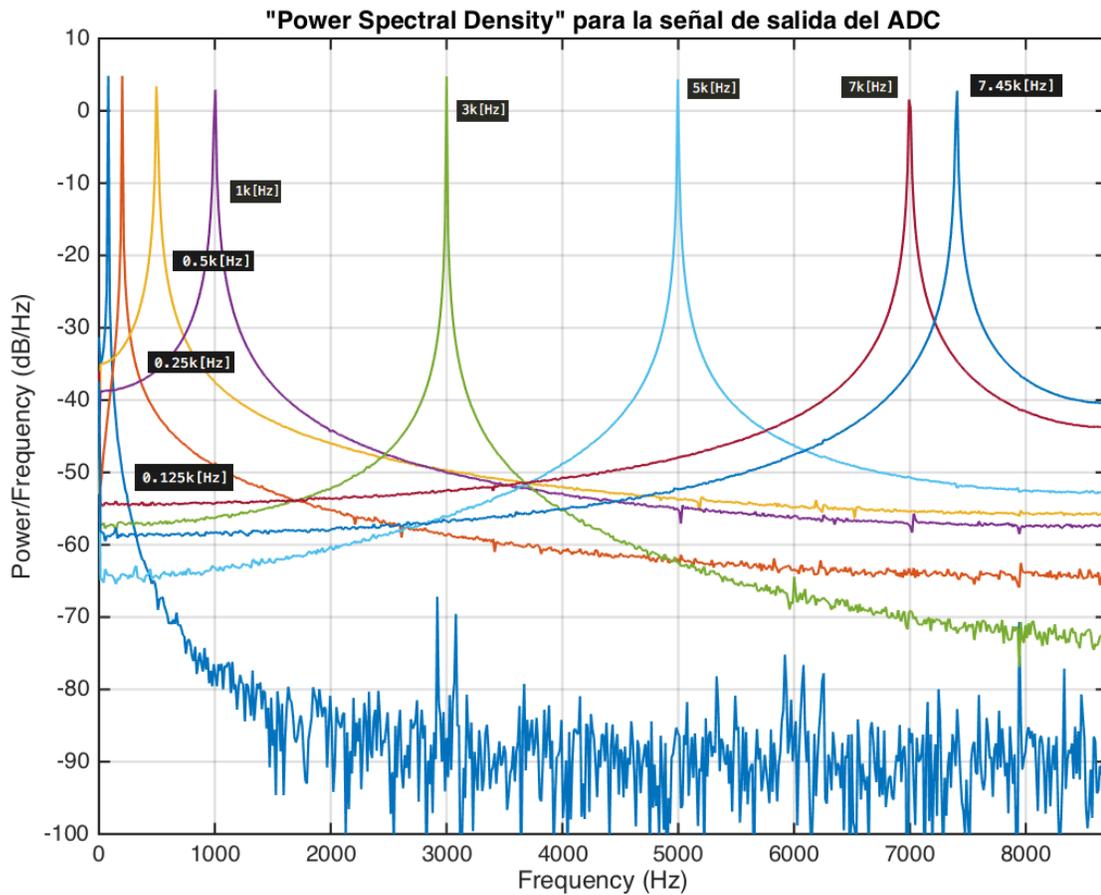


Figura 5.14: Gráfica conjunta del comportamiento de la función “Power Spectral Density” calculada para las mediciones de las ocho señales de salida del ADC (leídas con la tarjeta SBC AVR uc3-a3 Xplained e impresas desde el monitor serial “CuteCom”). Cada curva tiene un color distinto y se puede ver la componente fundamental en frecuencia de esa señal (0.08k[Hz], 0.2k[Hz], 0.5k[Hz], 1k[Hz], 3k[Hz], 5k[Hz] y 7k[Hz]). Además, se puede apreciar la aparición de una nueva curva con una frecuencia fundamental en 7.45k[Hz] aproximadamente. Esta curva es el resultado de la distorsión de la curva con una frecuencia fundamental en 10k[Hz] respectivamente. Esto es debido a que posee una frecuencia fundamental más altas que la frecuencia máxima de calibración (8.7k[Hz]).

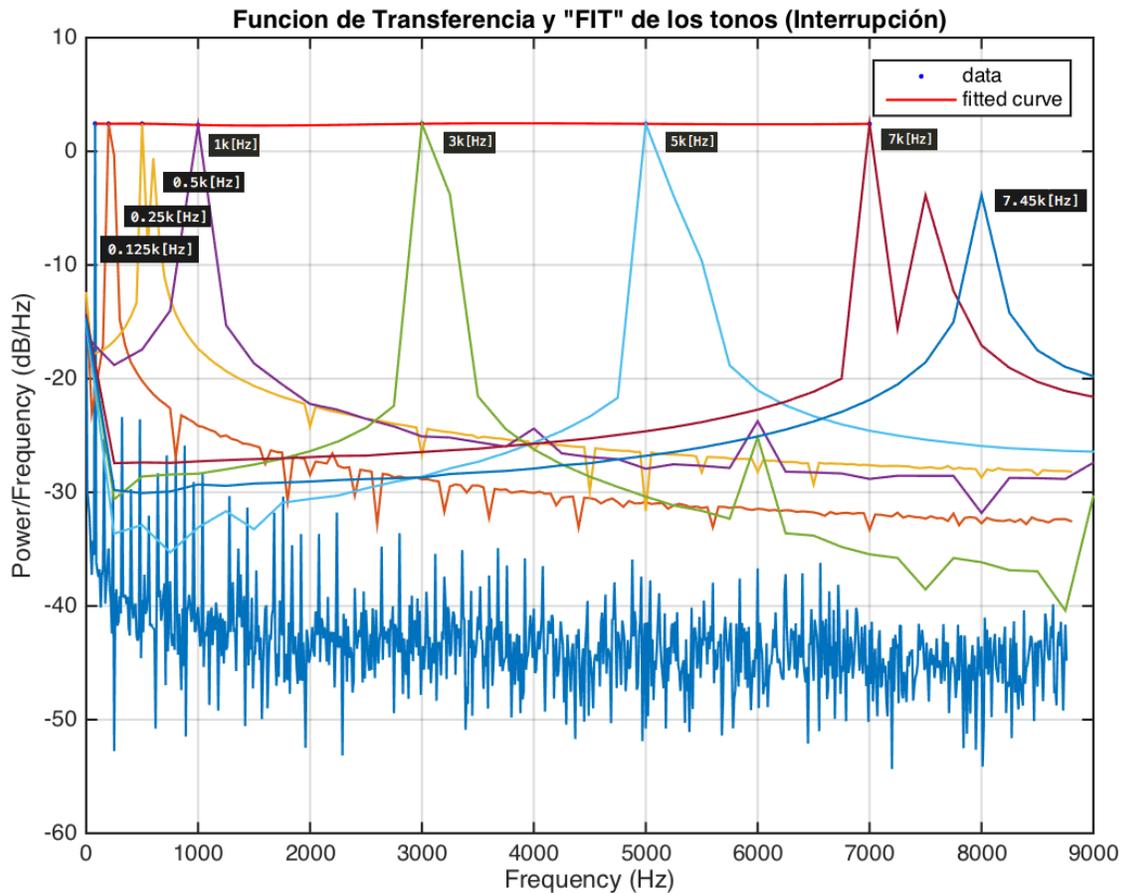


Figura 5.15: Gráfica conjunta del comportamiento de la “Función de Transferencia” calculada para las mediciones de las ocho señales de entrada y salida del ADC (leídas con el osciloscopio y la tarjeta SBC AVR uc3-a3 Xplained respectivamente). Cada curva tiene un color distinto y se puede ver la componente fundamental en frecuencia de esa señal (0.08k[Hz], 0.2k[Hz], 0.5k[Hz], 1k[Hz], 3k[Hz], 5k[Hz] y 7k[Hz]). Además, se puede apreciar la aparición de una nueva curva con una frecuencia fundamental en 7.45k[Hz] aproximadamente. Esta curva es el resultado de la alteración de la curva con una frecuencia fundamental en 10k[Hz] respectivamente. Esto es debido a que posee una frecuencia fundamental más altas que la frecuencia máxima de calibración (8.7k[Hz]). Por ultimo se realizó una interpolación (ajuste del tipo “cubicspline”) de las frecuencias fundamentales de todas las curvas que no sufrieron alteraciones de corrimiento en frecuencia.

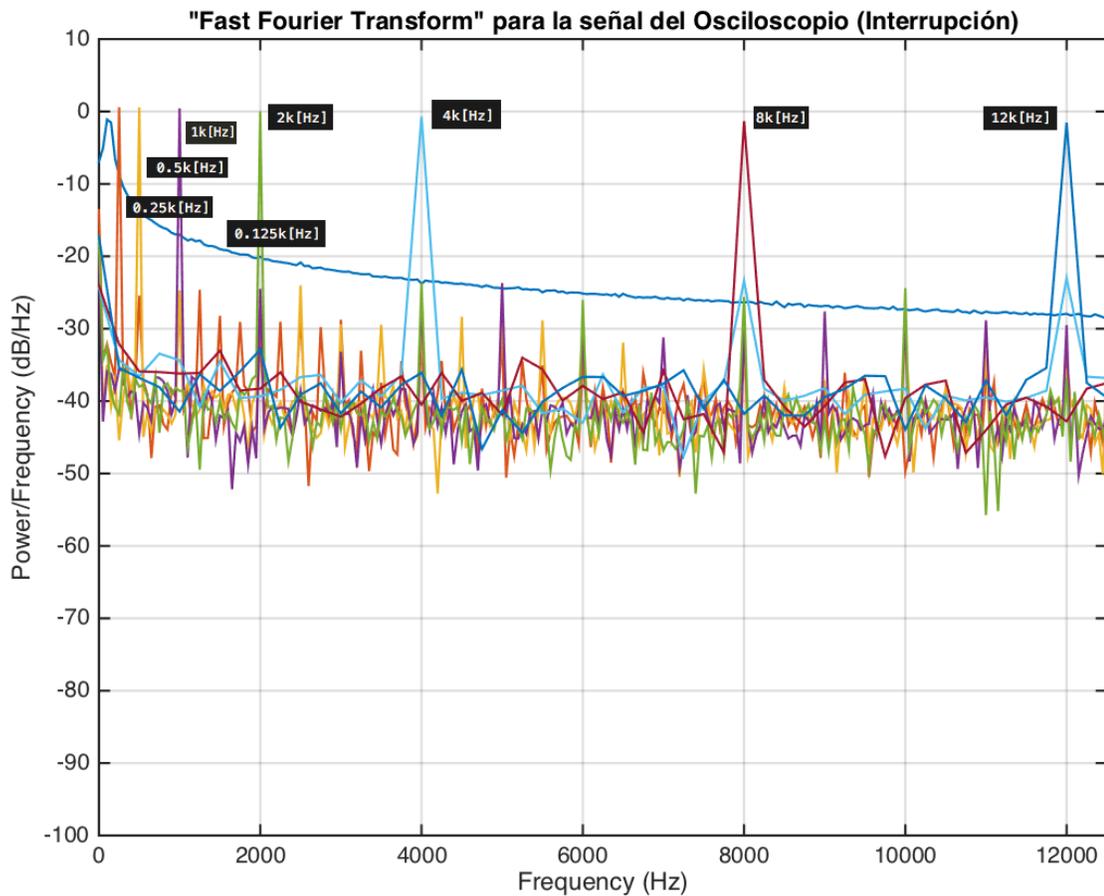


Figura 5.16: Gráfica conjunta del comportamiento de la función “Fast Fourier Transform” calculada para las mediciones de las ocho señales de entrada al ADC (leídas desde el osciloscopio), respecto al espectro de frecuencia. Cada curva tiene un color distinto y se puede ver la componente fundamental en frecuencia de esa señal (0.125k[Hz], 0.25k[Hz], 0.5k[Hz], 1k[Hz], 2k[Hz], 4k[Hz], 8k[Hz] y 12k[Hz]).

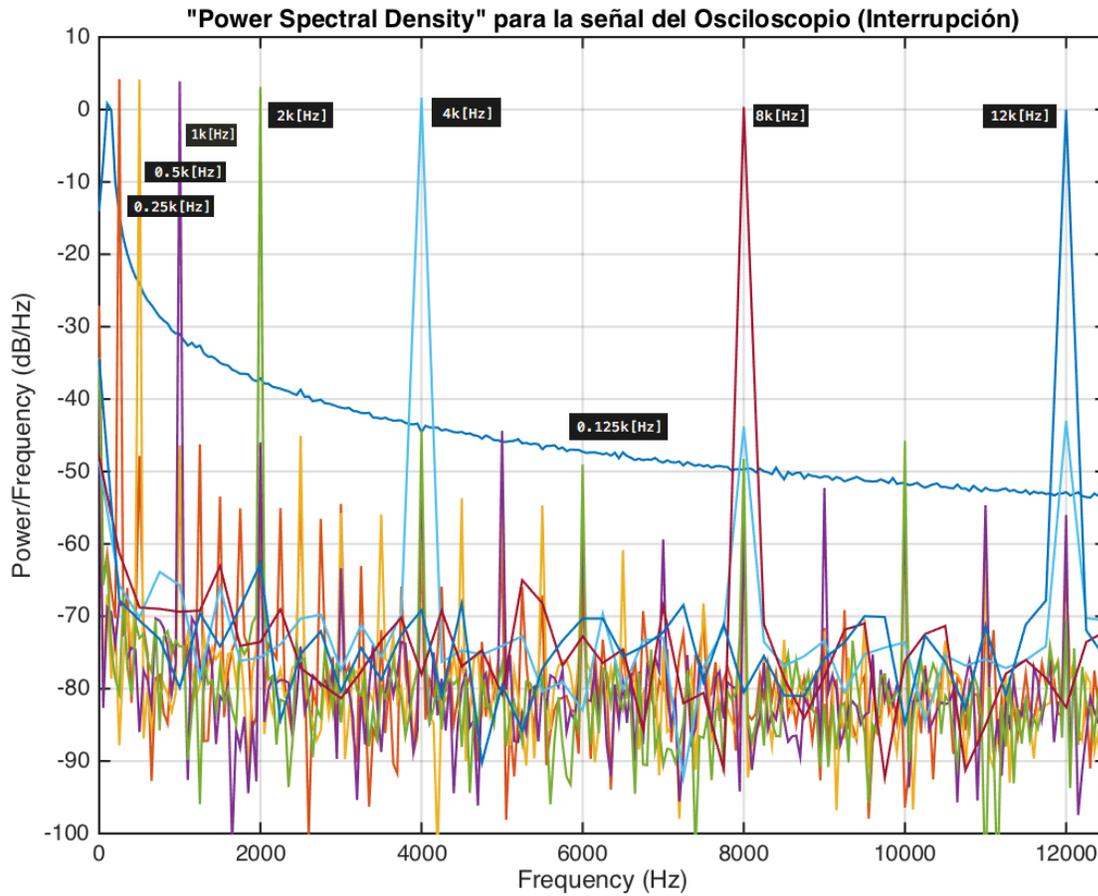


Figura 5.17: Gráfica conjunta del comportamiento de la función “Power Spectral Density” calculada para las mediciones de las ocho señales de entrada al ADC (leídas desde el osciloscopio), respecto al espectro de frecuencia. Cada curva tiene un color distinto y se puede ver la componente fundamental en frecuencia de esa señal (0.125k[Hz], 0.25k[Hz], 0.5k[Hz], 1k[Hz], 2k[Hz], 4k[Hz], 8k[Hz] y 12k[Hz]).

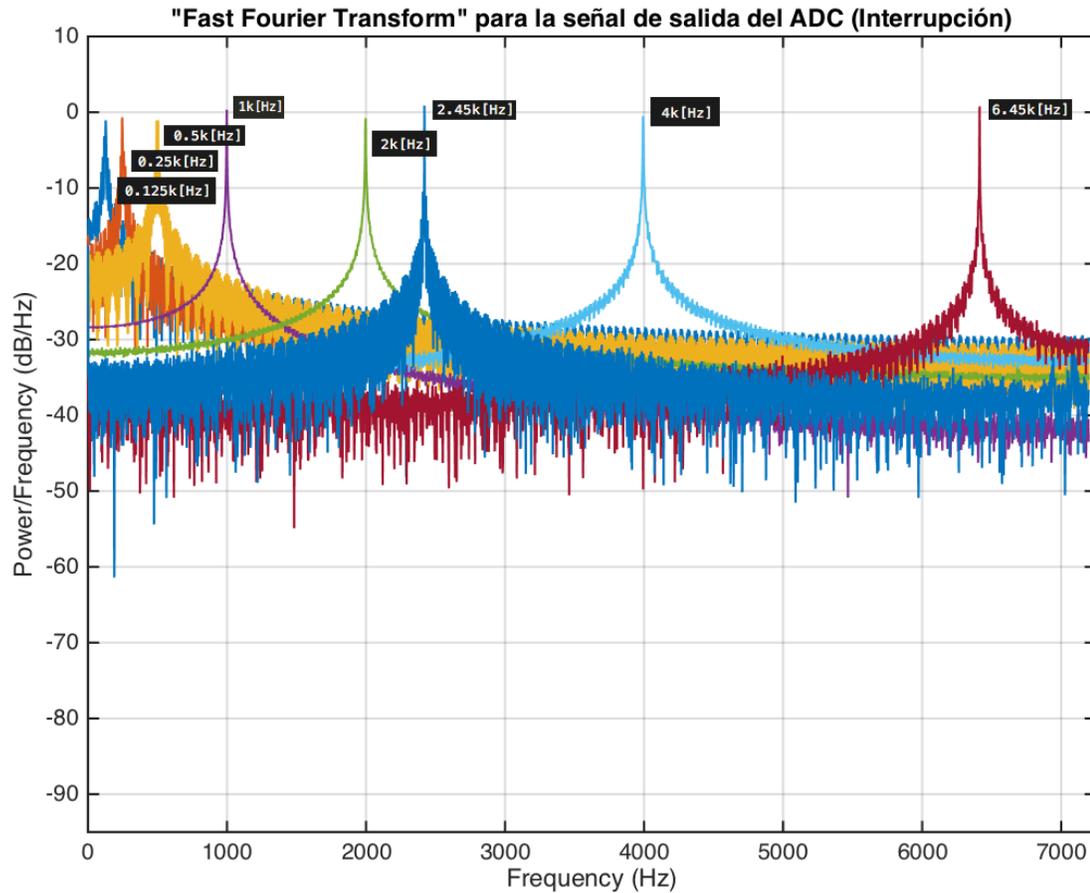


Figura 5.18: Gráfica conjunta del comportamiento de la función “Fast Fourier Transform” calculada para las mediciones de las ocho señales de salida del ADC (leídas con la tarjeta SBC AVR uc3-a3 Xplained e impresas desde el monitor serial “CuteCom”). Cada curva tiene un color distinto y se puede ver la componente fundamental en frecuencia de esas señales (0.125k[Hz], 0.25k[Hz], 0.5k[Hz], 1k[Hz], 2k[Hz], 4k[Hz]). Además, se puede apreciar la aparición de dos nuevas curvas con frecuencias fundamentales en 2.45k[Hz] y en 6.45k[Hz] aproximadamente. Estas dos curvas son el resultado de la distorsión de las curvas con frecuencias fundamentales en 12k[Hz] y 8k[Hz] respectivamente. Esto es debido a que poseen frecuencias fundamentales más altas que la frecuencia máxima de calibración (6k[Hz]).

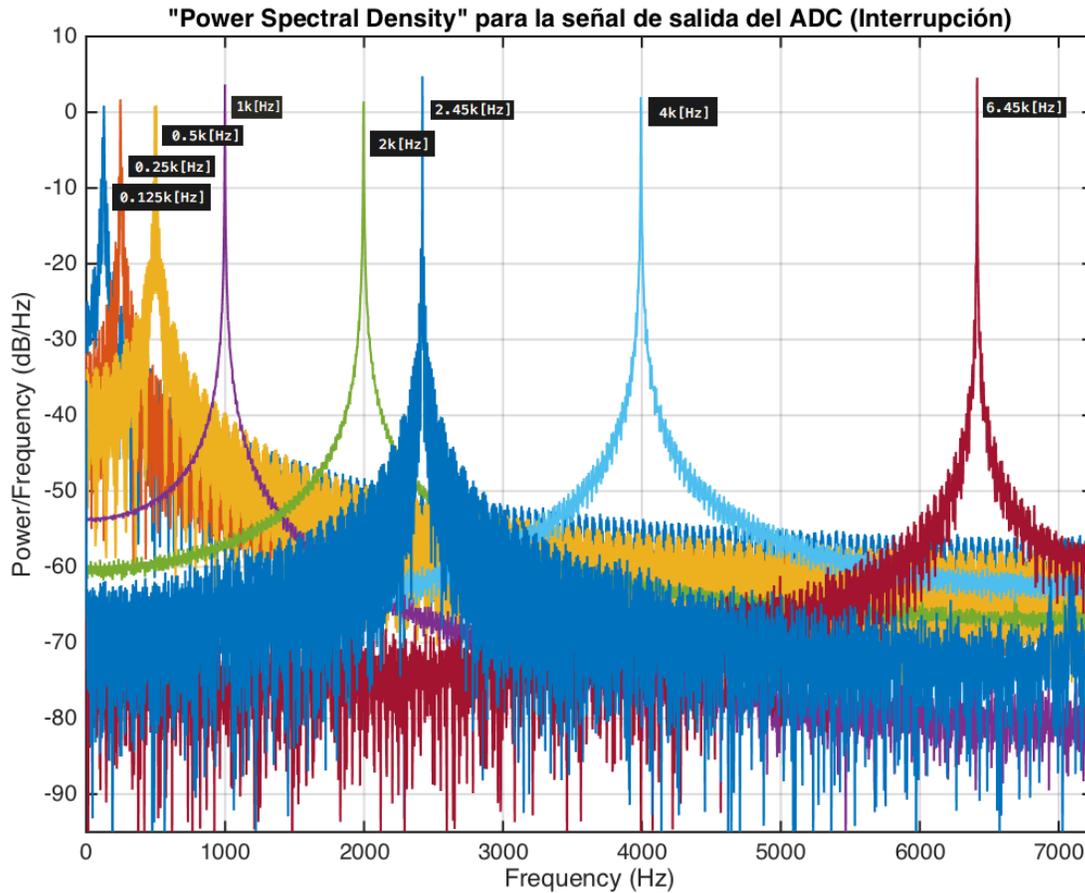


Figura 5.19: Gráfica conjunta del comportamiento de la función “Power Spectral Density” calculada para las mediciones de las ocho señales de salida del ADC (leídas con la tarjeta SBC AVR uc3-a3 Xplained e impresas desde el monitor serial “CuteCom”). Cada curva tiene un color distinto y se puede ver la componente fundamental en frecuencia de esas señales (0.125k[Hz], 0.25k[Hz], 0.5k[Hz], 1k[Hz], 2k[Hz], 4k[Hz]). Además, se puede apreciar la aparición de dos nuevas curvas con frecuencias fundamentales en 2.45k[Hz] y en 6.45k[Hz] aproximadamente. Estas dos curvas son el resultado de la distorsión de las curvas con frecuencias fundamentales en 12k[Hz] y 8k[Hz] respectivamente. Esto es debido a que poseen frecuencias fundamentales más altas que la frecuencia máxima de calibración (6k[Hz]).

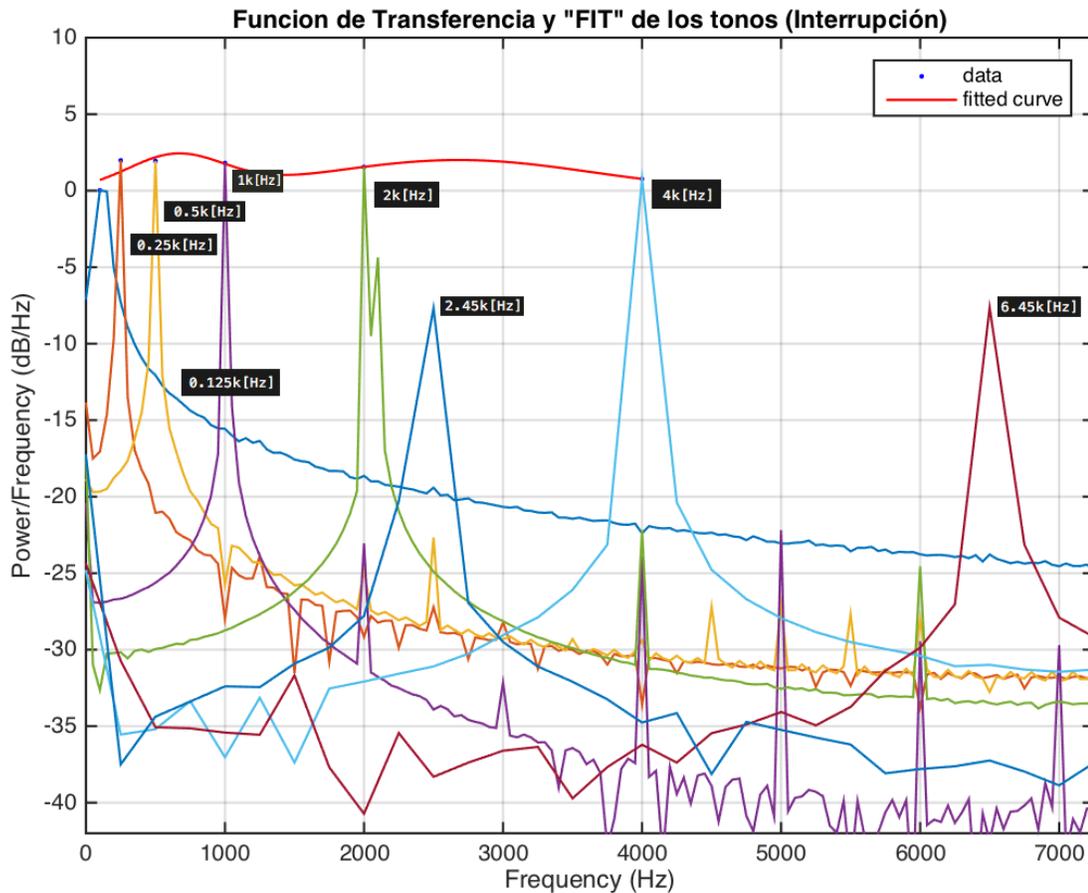


Figura 5.20: Gráfica conjunta del comportamiento de la “Función de Transferencia” calculada para las mediciones de las ocho señales de entrada y salida del ADC (leídas con el osciloscopio y la tarjeta SBC AVR uc3-a3 Xplained respectivamente). Cada curva tiene un color distinto y se puede ver la componente fundamental en frecuencia de esas señales (0.125k[Hz], 0.25k[Hz], 0.5k[Hz], 1k[Hz], 2k[Hz], 4k[Hz]). Además, se puede apreciar la aparición de dos nuevas curvas con frecuencias fundamentales en 2.45k[Hz] y en 6.45k[Hz] aproximadamente. Estas dos curvas son el resultado de la distorsión de las curvas con frecuencias fundamentales en 12k[Hz] y 8k[Hz] respectivamente. Esto es debido a que poseen frecuencias fundamentales más altas que la frecuencia máxima de calibración (6k[Hz]). Por último se realizó una interpolación (ajuste del tipo “cubicspline”) de las frecuencias fundamentales de todas las curvas que no sufrieron alteraciones de corrimiento en frecuencia.

Capítulo 6

Conclusiones

6.1. Conclusiones

En esta sección se detallarán las deducciones a las que se llegó una vez analizado los resultados anteriormente descritos. Además, se mencionan los posibles trabajos a futuro:

6.1.1. Calibración en valores de voltajes constantes: generar gráficos y comparar el desempeño frente al fabricante.

Se obtuvo mediciones para los “Errores Característicos” de calibración en valores de voltajes constantes de un ADC al analizar el comportamiento de los datos al ser graficados. Estos errores fueron muy cercanos a los mencionados en el datasheet del embebido, lo cual es esperable.

Si nos vamos al detalle, respecto al “Offset Error”, se obtuvo de manera empírica un valor igual a $0,0042[V]$. El fabricante nos entrega un valor máximo igual a $\pm 0,048\%FSR$. Sabiendo que el parámetro “Full Scale Input” es $\pm 10[V]$, se puede obtener la conversión a voltaje del “Offset Error” igual a $\pm 0,0048[V]$ [18]. Esto sugiere que estamos dentro de lo esperado.

Luego, respecto al “Gain Error” se obtuvo de manera empírica un valor igual a $0,1[V]$. El fabricante nos entrega un valor máximo igual a $\pm 0,8\%FSR$. Sabiendo que el parámetro “Full Scale Input” es $\pm 10[V]$, se puede obtener la conversión a voltaje del “Gain Error” igual a $\pm 0,08[V]$ [18]. Esto sugiere que estamos cerca de lo esperado, pero sobre el valor.

Respecto al error “Differential Non Linearity” (DNL), no se pudo obtener un valor estimado empírico, porque la diferencia de voltaje entre medida y medida fue cercana a $1[V]$, lo cual es muy por encima de la diferencia de voltaje que se necesita para estimar el tamaño empírico de un paso, también llamado “Least Significant Bits” (LSB). Por información del datasheet del ADC sabemos que el grosor (o tamaño) de un LSB para nuestra configuración es de $0,305m[V]$. El fabricante nos entrega un valor igual a $\pm 1LSB$ para el error DNL. Sa-

biendo que el parámetro “Maximum Input” es $10[V]$ y que el parámetro “Number of Bits” es 16, se puede obtener la conversión a voltaje del DNL, la cual es igual a $\pm 1,525 \cdot 10^{-4}[V]$ [18].

Respecto al error “Integral Non Linearity” (INL), no se pudo obtener un valor estimado empírico, porque no se pudo obtener la “Función de Transferencia” del sistema ADC para todo el espectro de frecuencia, ni tampoco se cuenta con la “Función de Transferencia” teórica entregada por el fabricante. El fabricante nos entrega un valor máximo igual a $\pm 3LSB$ para el error INL. Sabiendo que el parámetro “Maximum Input” es $10[V]$ y que el parámetro “Number of Bits” es 16, se puede obtener la conversión a voltaje del INL, la cual es igual a $\pm 4,577 \cdot 10^{-4}[V]$ [18].

Finalmente, respecto al “Total Unadjusted Error” (TUE) se puede inferir que, dada la información entregada por el fabricante, se puede calcular un valor teórico para este error siguiendo la ecuación:

$$\sqrt{(OffsetError[V])^2 + (GainError[V])^2 + (DNL[V])^2 + (INL[V])^2}$$

$$\sqrt{(0,0048[V])^2 + (0,08[V])^2 + (1,525 \cdot 10^{-4}[V])^2 + (4,577 \cdot 10^{-4}[V])^2}$$

$\sqrt{(0,0064)} = 0,0801[V]$ [18], para este ADC modelo AD7656 de Analog Device [16]. En conclusión, podemos inferir que el comportamiento del ADC al medir valores constantes es aceptable respecto a los estándares de la industria.

6.1.2. Primera calibración en frecuencia: organización del código y disminución de la tasa de sampleo de la conexión SPI.

Se alcanzó una mayor frecuencia de muestreo, llegando a obtener una Frecuencia de Nyquist de $8.7k[Hz]$. Esto fue gracias a realizar una organización de las acciones del software en sus distintas partes, que poseen distintos tiempos de ejecución. Primeramente, se obtuvo una sola lectura del ADC y se guardó en la memoria de la tarjeta AVR uc3-a3 Xplained. Posteriormente se quiso ocupar por completo el espacio en la memoria con lecturas. Finalmente se realizó la labor de enviar por la conexión SPI e imprimir el paquete de datos guardados por el monitor serial. Este procedimiento aislaba a la parte final la acción de “imprimir por el monitor serial”, la cual es notoriamente más lenta.

Además, es necesario mencionar que se eliminaron todas las instrucciones “delays” (intercaladas en el software). En un principio, estas eran necesarias para poder examinar (por ejemplo, con el osciloscopio) cada instrucción irremplazable del código, y lograr que funcionara correctamente.

Respecto al análisis posterior de los datos y la visualización de los gráficos, se pudo observar que las señales provenientes de la medición del osciloscopio poseían distinta cantidad de divisiones en el espectro de frecuencia (o el largo del periodo entre frecuencias). Mirando con mayor detalle, las señales provenientes de la medición del osciloscopio poseían la misma

cantidad de muestras, pero sus “ventanas de tiempo de muestreo” o el “largo de los experimentos” (nombrando como “experimento” a la toma de datos de una señal con una frecuencia fundamental fija) eran distintos para cada experimento (respecto al tiempo, en amplitud son todas iguales). Esto provoca que el periodo de muestreo cambie, por ende, la frecuencia de muestreo. Finalmente, esto provoca que el espectro en frecuencia de cada experimento este equidistante con un periodo propio y distinto al resto, lo cual cambia el detalle de la curva a observar. Algunos periodos que se observaron fueron: 10[hz], 50[hz] y 250[hz].

Es importante mencionar que los datos en su mayoría están alcanzados por la frecuencia de Nyquist (frecuencia máxima que es posible de samplear ocupando una Frecuencia de Muestreo determinada), por lo tanto, los experimentos son completamente muestreados con la Frecuencia de Muestreo que se utilizó (no hay efectos de “aliasing”). Solo un experimento quedó por encima de la frecuencia de Nyquist, este fue el que utilizó una señal sinusoidal con un tono en $10k[Hz]$. También se pudo observar la deformación en amplitud y desplazamiento de frecuencia de esta señal (probablemente efectos del “aliasing”) con mayor presencia al rededor de $7,45k[Hz]$, la cual se generó al tratar de samplear estos experimentos.

Se implementó el cálculo de la “Función de Transferencia” $|H(f)|^2 = PSD_Y(f)/PSD_X(f)$ (razón matemática entre la “Power Spectral Density” de salida y la de la entrada) [23] para poder comparar estas señales, y analizar como son procesadas en frecuencias por el ADC. En el caso de la primera calibración en frecuencia, se puede entregar la observación de que el sistema ADC amplifica 2.4 dB aproximadamente en la mayoría de los casos.

6.1.3. Segunda calibración en frecuencia: implementación de una ‘interrupción’ y disminución de la tasa de muestreo de la conexión SPI.

Se alcanzó una menor frecuencia de muestreo, llegando a obtener una Frecuencia de Nyquist de $6k[Hz]$. Esto fue debido a realizar la implementación de una interrupción del tipo “Timer/Counter”. Cuando se programó el “loop de la interrupción” y se manipuló su velocidad para aumentar la frecuencia de muestreo del ADC por la conexión SPI, las instrucciones de impresión en el “loop principal” del main, se vieron afectadas (se detuvieron) al ser interrumpidas por otros procesos del microcontrolador. En esta configuración del software, la prioridad de las impresiones por el monitor serial no es de las mayores del microprocesador. Se sospecha que las impresiones son interrumpidas por la propia interrupción “Timer/Counter”.

Al ver las funciones “Fast Fourier Transform” y “Power Spectral Density” de la salida del sistema ADC, y compararlas con las de la entrada, se puede observar que son relativamente parecidas en amplitud y forma, para las señales que tienen una frecuencia fundamental menor a la mitad de la Frecuencia de Muestreo de la experiencia. Si miramos las señales que están sobre, se ven alteradas por el sistema ADC.

Respecto al análisis posterior de los datos y la visualización de los gráficos, también se pudo observar que las señales provenientes de la medición del osciloscopio poseían distinta cantidad de divisiones en el espectro de frecuencia (o en el largo del periodo entre frecuencias).

Es importante mencionar que los datos en su mayoría están alcanzados por la frecuencia de Nyquist (frecuencia máxima que es posible de samplear ocupando una Frecuencia de Sampleo determinada), por lo tanto, los experimentos son completamente muestreados con la Frecuencia de Sampleo que se utilizó (no hay efectos de “aliasing”). Solo dos experimentos quedaron por encima de la frecuencia de Nyquist, estos fueron los que utilizaron señales sinusoidales con tonos en $8k[Hz]$ y en $12k[Hz]$. También se pudo observar la deformación en amplitud y desplazamiento de frecuencia de estas señales (probablemente efectos del aliasing) con mayor presencia al rededor de $2,45k[Hz]$ y $6,45k[Hz]$ respectivamente, las cuales se generaron al tratar de samplear estos experimentos.

Se implementó el cálculo de la “Función de Transferencia” $|H(f)|^2 = PSD_Y(f)/PSD_X(f)$ (razón matemática entre la “Power Spectral Density” de salida y la de la entrada) [23] para poder comparar estas señales, y analizar como son procesadas en frecuencias por el ADC. En el caso de la segunda calibración en frecuencia, no se puede entregar una observación muy acotada, porque el sistema ADC amplifica entre 1 y 2 dB en la mayoría de los casos.

Respecto a los gráficos se puede decir que revelan un comportamiento esperado para el funcionamiento de los valores de las señales en frecuencia ingresadas por el canal cinco de emisión/recepción del sistema ADC. Esto sugiere que podríamos esperar que este comportamiento sea el mismo para el resto de los canales de emisión/recepción.

Respecto a las exigencias sugeridas de la parte análoga, no se pudo lograr obtener la frecuencia máxima de muestreo de $21.2k[Hz]$. La máxima frecuencia de muestreo que se obtuvo fue de $17.5k[Hz]$, en el primer intento de calibración en frecuencia. Esta exigencia sugerida es necesaria para lograr muestrear todo el espectro de frecuencia que poseen las señales emitidas por la parte analógica del instrumento Langmuir. Sin embargo, se llegó a una frecuencia cercana, con la cual es posible recibir el 82,5 % de la información de la parte analógica.

El método que se utilizó para guardar las mediciones fue en la memoria interna del microprocesador. Esa memoria permite guardar un máximo de 1299 valores. Se piensa que esta cantidad de mediciones podría perjudicar la lectura de frecuencias muy bajas, que posean periodos más largos que la ventana de tiempo que se ocupó para muestrear los experimentos.

No se pudo obtener medidas empíricas para los “Errores Característicos” del ADC para la calibración en frecuencia, ya que no se contaba con los datos necesarios, ni el detalle correspondiente. Sin embargo, el fabricante entrega valores para cada uno de ellos. Para el error “Total Harmonic Distortion” (THD), el fabricante nos entrega un valor máximo igual a -90 dB. Para el error “Signal-to-Noise Ratio” (SNR), el fabricante nos entrega un valor típico igual a 88 dB. Para el error “Signal-to-Noise and Distortion” (SINAD), el fabricante nos entrega un valor típico igual a 88 dB. Por ultimo, para el error “Spurious Free Dynamic Range” (SFDR), el fabricante nos entrega un valor máximo igual a -100 dB.

6.2. Proposición de Trabajo a Futuro

Se proponen los siguientes trabajos a futuro para aumentar los alcances del payload.

Modificar el software para así terminar la implementación del guardado de datos en la tarjeta microSD que se encuentra ligada a la tarjeta que se conecta a la tarjeta SBC AVR uc3-a3 Xplained. Respecto al hardware el soporte de la tarjeta microSD esta completamente conectado al canal de comunicación SPI1. Respecto al software, hay que activar la conexión por el canal de comunicación SPI1 y posteriormente configurar a la tarjeta microSD para que se convierta en un lugar de almacenamiento para el microcontrolador.

Modificar el software y la configuración actual de la interrupción para alcanzar una frecuencia de muestreo mayor o igual a 21.2k[Hz], lo cual permitiría poder samplear por completo el ancho de banda de la señal de la parte analógica. Esto implicaría poder recibir el total de la información en frecuencia que en este momento está generando la parte análoga del instrumento Langmuir.

Modificar el software para aumentar la prioridad de las impresiones por el monitor serial para que así no sean interrumpida por procesos internos del microcontrolador o por la misma interrupción del tipo “Timer/Counter”. Además, como una tarea alternativa se podría cambiar el destino de estas impresiones, enviarlas directamente a almacenar en la tarjeta microSD y posteriormente, confirmando que fueron bien guardados los datos, evaluar la velocidad de lectura de muestras, analizando los canales de comunicación SPI con un osciloscopio. Esto ultimo con el objetivo de poder seguir aumentando la velocidad de la interrupción del tipo “Timer/Counter” y que las impresiones por el monitor serial dejen de ser el gran impedimento.

Para obtener una caracterización más robusta y detallada de los efectos que experimentan las señales en frecuencia al atravesar el sistema ADC, se propone aumentar las mediciones de señales sinusoidales con una frecuencia característica para obtener una curva de interpolación más detallada. Otra forma de caracterizar los efectos que experimentan las señales en frecuencia, es utilizar una señal de ruido blanco. Esta señal posee un espectro en frecuencia característico y el espectro de su “Power Spectral Density” es continua. Por lo tanto, se vería información útil en más lugares en frecuencia para la función de transferencia del sistema ADC.

Ocupar un osciloscopio que sea capaz de mantener constante su ventana de tiempo al samplear una señal. Esto seria de gran ayuda para calcular y comparar espectros de frecuencia de señales de entrada a un sistema a estudiar y no perder resolución en el detalle de las frecuencias equidistantes que conforman a la “Fast Fourier Transform”. En el caso particular de este informe, esa característica habría sido útil para calcular la “Power Spectral Density” de las señales de entrada y la Función de Transferencia del sistema.

Como objetivo final deseado a largo plazo, es lograr posicionar todos los elementos del payload -parte analógica, parte digital y la conexión con el resto del satélite (la cual incluye la alimentación de energía)- dentro de un nuevo diseño de la tarjeta del instrumento Langmuir. En la cual se desea reordenando sus componentes embebidos para que la tarjeta ocupe el

menor espacio posible y pueda ser uno de los payload incluidos en los nanosatélites de futuras misiones.

Capítulo 7

Bibliografía

- [1] Cubesat. <http://www.cubesat.org/>.
- [2] Definition of ionosphere. <https://www.merriam-webster.com/dictionary/ionosphere>.
- [3] Definition of plasma. <https://www-istp.gsfc.nasa.gov/Education/Mplasma.html>.
- [4] International reference ionosphere. <https://iri.gsfc.nasa.gov/>.
- [5] Nanosatellite and cubesat database. <https://www.nanosats.eu/>.
- [6] Suchai flight software. <https://github.com/spel-uchile/SUCHAI-Flight-Software>.
- [7] Projects. <https://spel.ing.uchile.cl/projects.html>, 2019.
- [8] Space and planetary exploration laboratory. <http://spel.ing.uchile.cl/>, 2019.
- [9] Suchai-i. <http://spel.ing.uchile.cl/suchai.html>, 2019.
- [10] Atmel. Quick start guide for spi driver on avr32 devices. http://asf.atmel.com/docs/latest/common.components.memory.sd_mmc.example1.evk1100/html/avr32_drivers_spi_quick_start.html.
- [11] Atmel. Atmel avr32918: Uc3-a3 explained hardware user's guide. <https://www.microchip.com/>, Jul, 2012.
- [12] Aroh Barjatya. *Langmuir Probe Measurements In The Ionosphere*. Utah State University, Utah, USA, May, 2007.
- [13] Atmel Corporation. Timer/counter example 3. https://github.com/eewiki/asf/blob/master/avr32/drivers/tc/example3/tc_example3.c, 2015.
- [14] L. H. Brace D. Bilitza, W. R. Hoegy and R. F. Theis. *Evaluation of the International Reference Ionosphere with the large AE-C and DE2 databases*. vol. 8, no. 4, p. 209

- edition, 1988.
- [15] Seth Foote Daniel McClure and Hank Voss. *TSAT Solid State Detector and Plasma Probe Particle Detectors*. Taylor University, USA, 2013.
 - [16] Analog Devices. *Data Sheet: AD7656-1/AD7657-1/AD7658-1*. USA, 2012.
 - [17] Marcos Díaz. Entrevista para el fondo nacional de desarrollo científico y tecnológico. <http://www.conicyt.cl/fondecyt/2017/07/18/suchai-desarrollo-espacial-chileno-de-la-mano-de-conicyt/>, 2017.
 - [18] Texas Instrument. Adc performance parameters - convert the units correctly! <https://www.ti.com>, May, 2013.
 - [19] K. Hirao Koichiro Oyama. *Inaccuracies in electron density estimates due to surface contamination of Langmuir probes*. *Planetary and Space Science*. p. 87-89 edition, 1976.
 - [20] Miguel Martínez Ledesma. *Planar Langmuir Probe Circuit Simulation*. Universidad de Chile, Santiago, CHILE, 1st edition, 1 March, 2017.
 - [21] Miguel Martínez Ledesma. *Langmuir Probe - New Design Dec2017 - vfinal3*. Universidad de Chile, Santiago, Chile, Dec, 2017.
 - [22] Harry Nyquist. *Certain Topics in Telegraph Transmission Theory*. Transactions of the A. I. E. E, New York, USA, Feb, 1928.
 - [23] Alan V. Oppenheim and George C. Verghese. Signals, systems and inference. introduction to communication, control and signal processing. https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-011-introduction-to-communication-control-and-signal-processing-spring-2010/readings/MIT6_011S10_notes.pdf, Spring, 2010.
 - [24] Koichiro Oyama. *DC Langmuir Probe for Measurement of Space Plasma: A Brief Review*. 2015.
 - [25] C. T. Steigies and A. Barjatya. *Contamination effects on fixed-bias Langmuir probes*. Germany and USA, 2012.
 - [26] U.-P. Hoppe¹ A. Pedersen¹ J. I. Moen¹ M. Friedrich³ M. Rapp¹ T. A. Bekkeng, A. Barjatya. *Payload charging events in the mesosphere and their impact on Langmuir type electric probes*. 2013.

Anexos A

A.1. ADC Parameters Unit Conversion.pdf

ADC Performance Parameters - Convert the Units Correctly!

Shridhar Atmaram More
Digital/Analog Converters

ABSTRACT

All semiconductor manufacturers define operation and performance of Analog-to-Digital Converters (ADC) using common specifications. Typically, different manufacturers use different units to define the same ADC specification. This inconsistent use of units may result in misinterpretation of performance while comparing various ADCs, possibly leading to improper selection of the ADC. This application report explains how to convert the specification numbers between various units.

Contents

1	Introduction	2
2	DC/Static Specifications	2
	2.1 Offset Error	2
	2.2 Gain Error	3
	2.3 Differential Non Linearity (DNL)	4
	2.4 Integral Non Linearity (INL)	4
	2.5 Total Unadjusted Error (TUE)	5
3	AC (or Dynamic) Errors	5
	3.1 Total Harmonic Distortion (THD)	6
	3.2 Signal-to-Noise Ratio (SNR)	6
	3.3 Signal-to-Noise and Distortion (SINAD)	6
	3.4 Spurious Free Dynamic Range (SFDR)	6
	3.5 Conversion Between Units	7
4	References	7

List of Figures

1	Positive Offset Error	2
2	Negative Offset Error	2
3	Gain Error	3
4	DNL	4
5	INL	5
6	Typical ADC Output Spectrum Plot.....	6

1 Introduction

The important ADC parameters are usually divided in two broad categories: DC (or Static) Input Specifications and AC (or Dynamic) Input Specifications. This report covers the following specifications in detail.

1. DC (or Static) Input Specifications:
 - (a) Offset Error and drift
 - (b) Gain Error and drift
 - (c) Differential Non Linearity (DNL)
 - (d) Integral Non Linearity (INL)
 - (e) Total Unadjusted Error (TUE)
2. AC (or Dynamic) Input Specifications:
 - (a) Total Harmonic Distortion (THD)
 - (b) Signal to Noise Ratio (SNR)
 - (c) Signal to Noise and Distortion (SINAD)
 - (d) Spurious Free Dynamic Range (SFDR)

For all the calculations shown throughout this document, the following typical specifications are used for the ADC:

- Resolution = 16 bit
- Unipolar Single-ended configuration
- $V_{ref} = 5\text{ V}$
- Full scale input = 5 V

The calculation methodology can be applied to Differential and Bipolar ADCs as well.

2 DC/Static Specifications

The DC specifications for an ADC provide an understanding of the device behavior for dc or a very-low-frequency input signal. Important DC specifications are explored in the following sections.

2.1 Offset Error

Offset Error is also referred to as *zero scale error*. It is defined as the difference between Ideal offset point (0.5 LSB) to actual offset and it can be a positive or negative value (as shown in [Figure 1](#) and [Figure 2](#)).

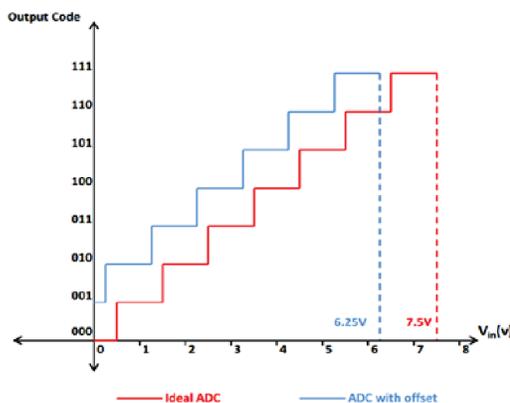


Figure 1. Positive Offset Error

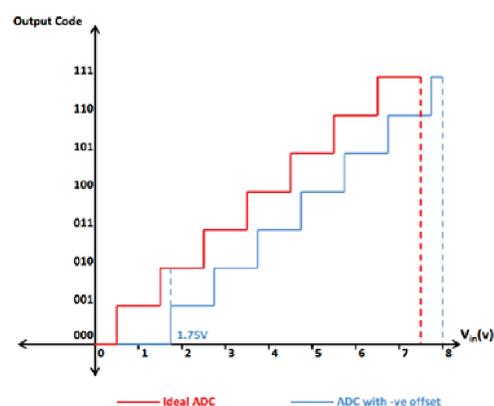


Figure 2. Negative Offset Error

Offset Error value is usually specified using one of the following units: Volts, Least Significant Bits (LSB), %Full Scale Value (%FSV), and parts per million (ppm). FSV is sometimes also referred to as FSR (Full Scale Range). Parts per million (ppm) is usually associated with the actual unit, for example, 1 ppm of 1 V = 1 μ V.

Each of these can be converted to another form by using equations and formulas.

For the above example, you can convert between different units as shown in the following example.

Calculate a 3 LSB offset error conversion to Volts:

$$\text{Offset Error (V)} = \text{Error in LSB} \times \text{Maximum Input} / (2^{\text{number of bits}})$$

$$\text{Offset Error (V)} = 3 \times 5 \text{ V} / (2^{16})$$

$$\text{Offset Error (V)} = 0.000229, \text{ that is, } 229 \mu\text{V}$$

$$\text{Offset Error (\%FSV)} = \text{Offset Error (V)} \times 100 / \text{Full scale value}$$

$$\text{Offset Error (\%FSV)} = 0.00458\%$$

ppm, with regard to full scale voltage, is Offset Error (ppm FSV) = 46 ppm, in the example above.

Though the offset error is usually specified at 25°C in the data sheets, the offset does vary with temperature. The variation in offset is specified as *Offset Drift* and denoted as ppm/°C. The actual offset at any temperature can be calculated by adding the drift to offset value calculated for room temperature.

For the above example if the drift is specified as 1 ppm/°C of REF V.

$$\text{Offset at } 85^\circ\text{C} = 229 \mu\text{V} + [(85 - 25) \times 5 \mu\text{V}] = 529 \mu\text{V}.$$

2.2 Gain Error

Gain Error is also referred to as *Full Scale Error*. It represents the difference between ideal voltage which provides Full scale output code (in our example 0xFFFF) versus the actual voltage for which the converter provides full scale output code. This measurement is done after calibrating the ADC readings for the offset error. The error represents the difference in the ideal and actual slopes; therefore, the percentage error is same across all the ADC steps (Figure 3).

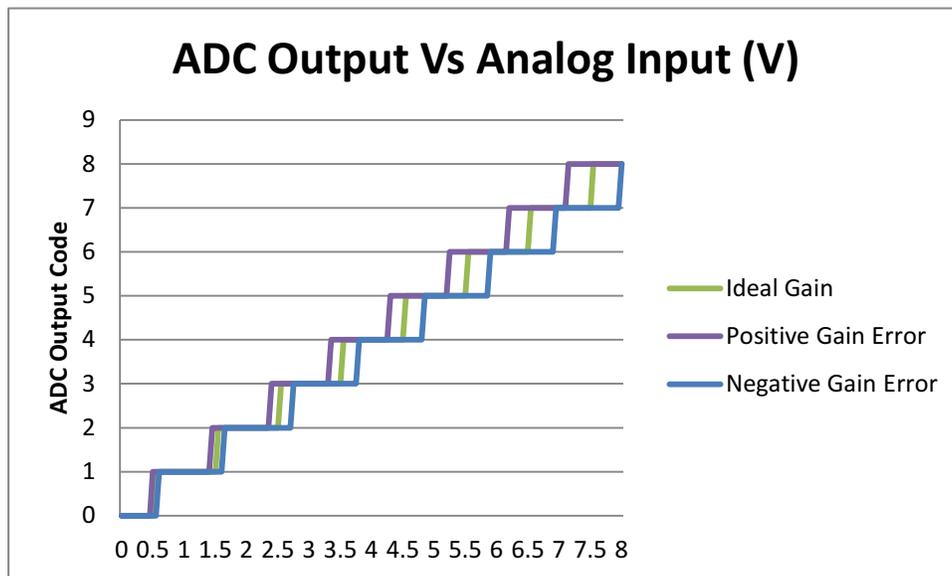


Figure 3. Gain Error

Gain Error value is usually specified using one of the following units: %FSV or LSB or Volts.

For the above ADC, if the gain error is 4 LSB, then it can be converted to Volts as follows:

- Gain Error (Volts) = Error in LSB × Maximum Input / ($2^{\text{number of bits}}$)
- Gain Error (Volts) = $4 \times 5 / (2^{16}) = 0.000305 \text{ V}$, that is, 305 μV

This means the ADC will reach 0xFFFF code for input voltage of 4.999656 V.

If the gain error is -4 LSB, then the device will reach 0xFFFF code for input voltage 5.000267 V.

- Gain Error (%FSV) = Gain Error (V) × 100 / Full scale value
- Gain Error (%FSV) = 0.0061%

Similar to offset error, the gain error is usually specified at 25°C in the data sheets and the gain also varies with temperature. The variation in gain is specified as *Gain Drift* and denoted as ppm/°C. The actual gain error at any temperature can be calculated by adding the drift to gain error value calculated for room temperature.

For the above example, if the drift is specified as 1 ppm/°C of REF V.

$$\text{Gain Error at } 85^\circ\text{C} = 305 \mu\text{V} + [(85 - 25) \times 5 \mu\text{V}] = 605 \mu\text{V}.$$

2.3 Differential Non Linearity (DNL)

Differential Non Linearity, also referred as Differential Linearity Error, describes deviation between ideal step-size to actual step-size observed for each ADC code. The ideal step-size is 1 LSB. A typical DNL curve is as shown in [Figure 4](#).

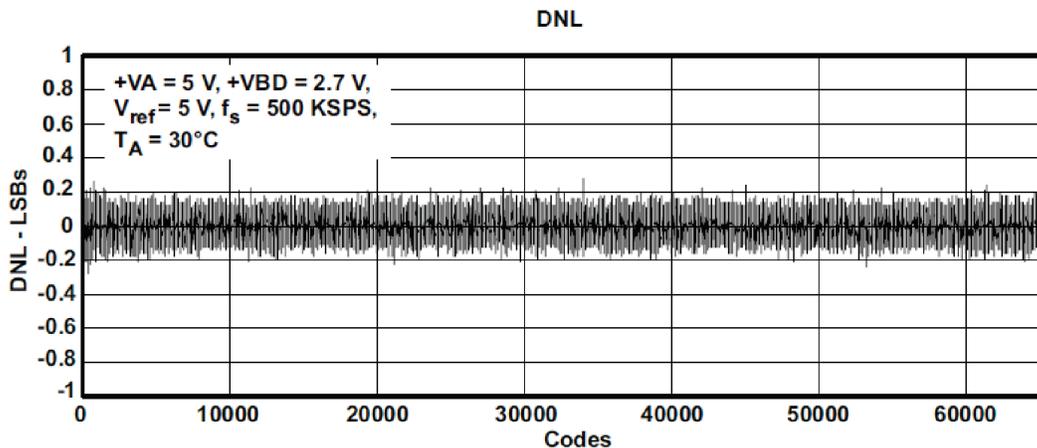


Figure 4. DNL

DNL value is usually specified using one of the following units: LSB or %FSV

We can use the same equations explained in Offset and Gain Error sections to convert from LSB to %FSV.

For the above ADC, 3 LSB error = 0.0046% FSV

2.4 Integral Non Linearity (INL)

Integral Non Linearity, also referred as Integral Linearity Error, describes the deviation of the actual transfer function with respect to ideal transfer function for an ADC. By definition, INL for a particular code is the summation of DNL array till that code. A typical INL curve is as shown in [Figure 5](#).

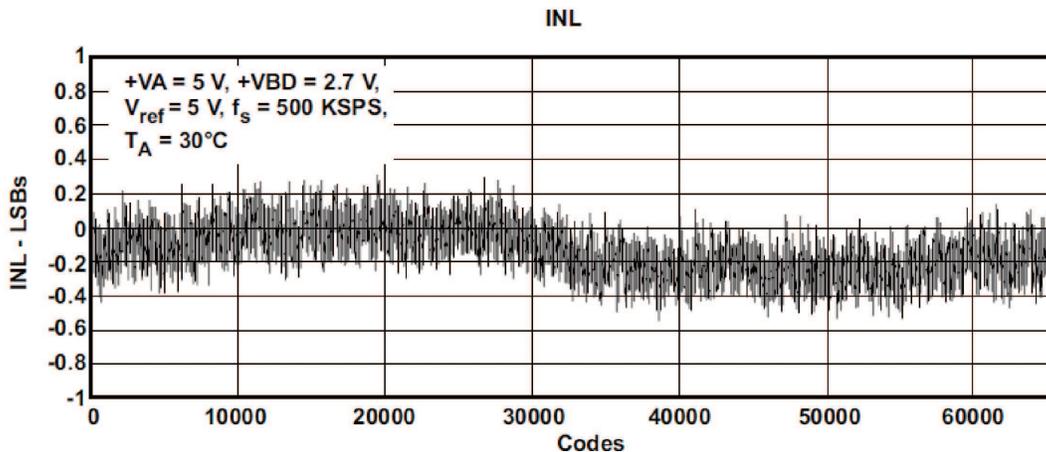


Figure 5. INL

INL value is usually specified using one of the following units: LSB or %FSR.

The equations explained in [Offset Error](#) and [Gain Error](#) sections can be used to convert between LSB to %FSV.

For the example ADC, 3 LSB error = 0.0046% FSV.

2.5 Total Unadjusted Error (TUE)

The Total Unadjusted Error (TUE) specification is an indication of the ADC's worst rms error without applying any Offset or Gain Error correction. The TUE number is not calculated as a summation of Offset, Gain, DNL and INL errors.

Since it is an RMS number, the TUE is calculated as

$$\text{TUE} = \sqrt{\text{sq}(\text{Offset Error}) + \text{sq}(\text{Gain Error}) + \text{sq}(\text{DNL}) + \text{sq}(\text{INL})}$$

It is important to convert all the errors to same units.

For example, ADC with Offset Error = 3 LSB, Gain Error = 4 LSB, DNL = 1 LSB and INL = 2 LSB, will have

$$\text{TUE} = \sqrt{9 + 16 + 1 + 4} \text{ and } \text{TUE} = 5.48 \text{ LSB}$$

Since the offset and gain error can be calibrated out from the ADC transfer curve, the actual error in the application will be dominated by INL and DNL errors.

3 AC (or Dynamic) Errors

An ADC's performance has different important specifications when the input varies quickly. These different parameters which define the ADC performance with Dynamic input are mostly specified using single input frequency. The ADC output array is processed using FFT and analyzed for dynamic specifications. Each specification is usually associated with input signal specs in terms of frequency and amplitude.

A typical ADC output spectrum plot is shown in [Figure 6](#).

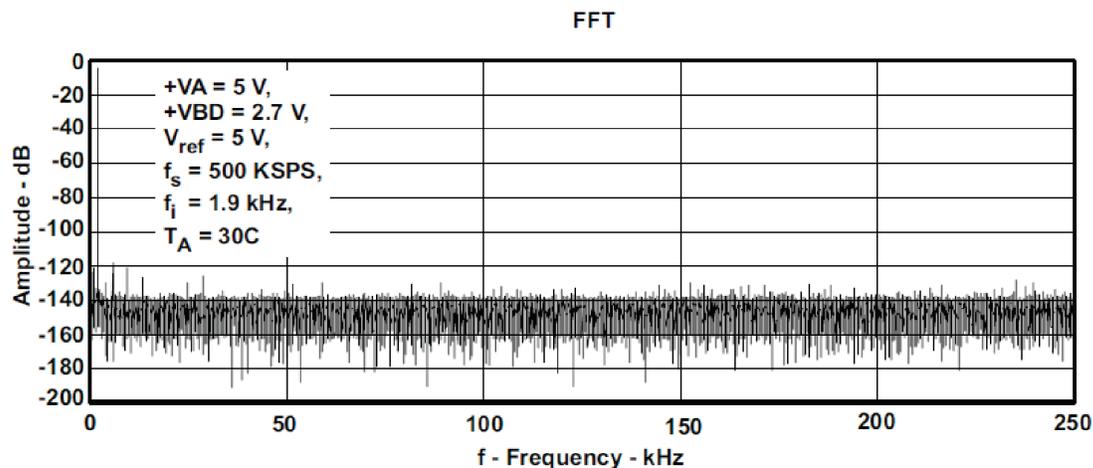


Figure 6. Typical ADC Output Spectrum Plot

3.1 Total Harmonic Distortion (THD)

The Total Harmonic Distortion (THD) specification provides information regarding the harmonic energy present in the frequency spectrum for a particular input frequency. The frequency spectrum is typically shown till the Nyquist frequency and the THD calculation usually takes into account all the harmonics energy till Nyquist. Harmonics beyond Nyquist fall back into the frequency spectrum as noise or spurious tone. These are taken care of in the SNR and SINAD specifications.

The parameter is usually specified in terms of dB or %.

THD = Summation of harmonic energy / Fundamental input energy

3.2 Signal-to-Noise Ratio (SNR)

The Signal-to-Noise Ratio (SNR) specification provides information regarding the noise energy excluding the fundamental and harmonic energy present in the frequency spectrum for a particular input frequency. The SNR calculation usually integrates noise till Nyquist frequency. If not, the specifications will imply the band of frequency where the noise is integrated.

The parameter is usually specified in terms of dB, V_{rms} , or %.

SNR = Fundamental input energy / Summation of noise energy

3.3 Signal-to-Noise and Distortion (SINAD)

The Signal-to-Noise and Distortion (SINAD) specification provides information regarding the noise and harmonic energy present in the frequency spectrum.

The parameter is usually specified in terms of dB, V_{rms} , or %.

SINAD = Fundamental input energy / Summation of noise + distortion energy

3.4 Spurious Free Dynamic Range (SFDR)

The Spurious Free Dynamic Range (SFDR) specification provides information regarding the difference between maximum amplitude tone in frequency spectrum and the fundamental input tone.

The parameter is usually defined in dB.

SFDR = Fundamental input energy – Max (all frequency bins except fundamental)

3.5 Conversion Between Units

- Percentage to dB
 - $\text{dB} = 20 \times \log (\text{Percentage} / 100)$
 - that is, 1% = -40 dB and 0.1% = -60 dB
- dB to Vrms
 - Assuming input amplitude as 1 Vrms and for 60 dB SNR,
 - Noise Amplitude is $\text{dB} = 20 \times \log (\text{Input Amplitude} / \text{Noise Amplitude})$
 - Noise Amplitude Vrms = 0.001, that is, 1 mVrms
 - If the spec is -60 dB, then the formula will be $\text{dB} = 20 \times \log (\text{Noise Amplitude} / \text{Input Amplitude})$

4 References

1. *Understanding Data Converters* – [SLAA013](#)
2. ADS8318 data sheet – [SLAS568A](#)

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as "components") are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI's terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers' products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers' products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI's goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or "enhanced plastic" are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have **not** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products

Audio	www.ti.com/audio
Amplifiers	amplifier.ti.com
Data Converters	dataconverter.ti.com
DLP® Products	www.dlp.com
DSP	dsp.ti.com
Clocks and Timers	www.ti.com/clocks
Interface	interface.ti.com
Logic	logic.ti.com
Power Mgmt	power.ti.com
Microcontrollers	microcontroller.ti.com
RFID	www.ti-rfid.com
OMAP Applications Processors	www.ti.com/omap
Wireless Connectivity	www.ti.com/wirelessconnectivity

Applications

Automotive and Transportation	www.ti.com/automotive
Communications and Telecom	www.ti.com/communications
Computers and Peripherals	www.ti.com/computers
Consumer Electronics	www.ti.com/consumer-apps
Energy and Lighting	www.ti.com/energy
Industrial	www.ti.com/industrial
Medical	www.ti.com/medical
Security	www.ti.com/security
Space, Avionics and Defense	www.ti.com/space-avionics-defense
Video and Imaging	www.ti.com/video

TI E2E Community

e2e.ti.com

A.2. Planar Langmuir Probe - Circuit Simulation - v1.pdf

Planar Langmuir Probe

Circuit Simulation

Author:	Miguel Martínez Ledesma	Email:	miguel.martinez@ing.uchile.cl
Date:	1 March 2017	Version:	1

INDEX

1.	Introduction	2
2.	Trans-Impedance Amplifier	3
2.1.	Design	3
2.2.	Transfer Function	3
2.3.	Frequency Response	4
3.	Complete Circuit.....	6
3.1.	Design	6
3.2.	Transfer Function	6
3.3.	Frequency Response	7
3.4.	Consumption	8
3.5.	Noise.....	8
4.	Temperature Circuit	10
4.1.	Design	10
4.2.	Transfer Function	10
4.3.	Frequency Response	11
5.	Annex - Schematic.....	12

1. Introduction

During February 2017 Dr. Marcos Díaz and PhD candidate Miguel Martínez-Ledesma visited the Space and Atmospheric Instrumentation Laboratory (SAIL) of the Embry Riddle Aeronautical University (ERAU) at Daytona Beach, Florida, EEUU. This visit was intended to test the Langmuir Probe (LP) sensor design that we created following the recommendations indicated by Dr. Aroh Barjatya.

After several design changes we were able to fabricate our first LP circuit and verify the analog circuitry. An image of the circuit is shown in Figure 1, and the schematic of it is included in the Annex.

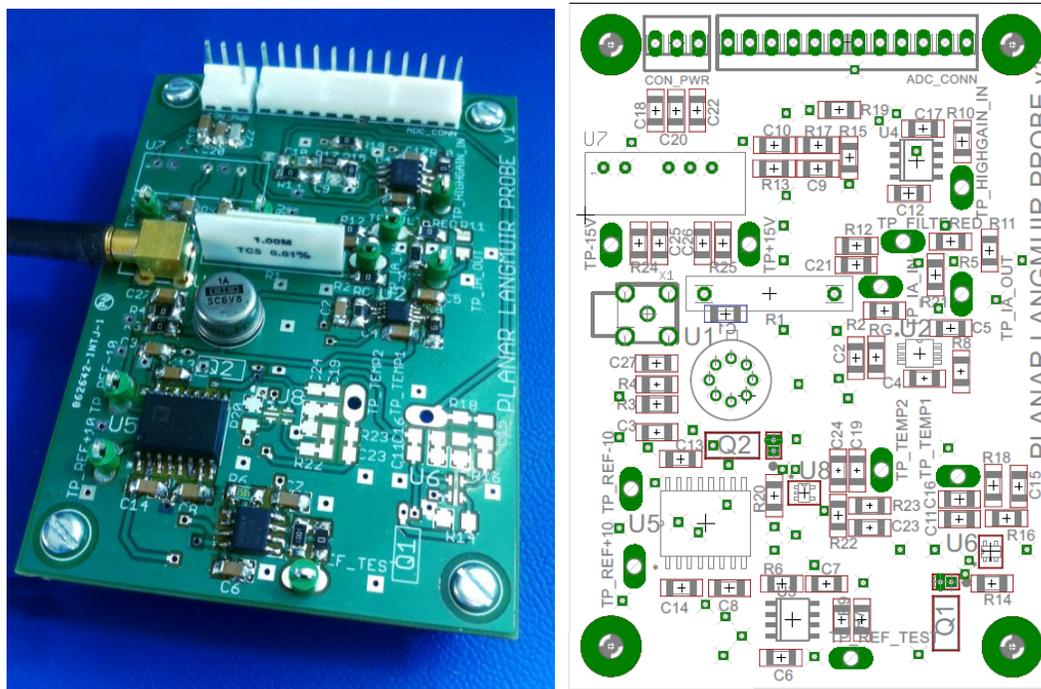


Figure 1. (Left) Image of the LP sensor circuit developed and (Right) location of electronic components on the circuit

Initial tests indicate that the LP sensor circuit developed is able to measure correctly electric currents that are collected by a planar electrode located on the spacecraft surface pointing to the ram direction of the satellite. Nevertheless, these tests indicate that the circuit was not responding exactly as initially designed. Electronic components errors, tolerances, and offsets modified slightly the circuit behavior.

Several simulations have been done in order to estimate the real effects on the circuit response of those errors. The simulations have been done using the PSpice simulator software of the OrCAD 16.6 package. Spice modules of the electronic components have been downloaded from manufacturer web page and have been introduced into the PSpice simulator environment.

2. Trans-Impedance Amplifier

2.1. Design

The Trans-Impedance Amplifier (TIA) is the circuit responsible of converting the small input current of the probe into an output voltage. This is done by using a very low input bias current operational amplifier (op-amp) and a feedback resistor. The amplification factor is set by the value of the feedback resistor depending on the range of input current desired to measure.

The circuit simulated is shown in Figure 2. In our design the operational amplifier model OPA128 has only 75fA bias current and the feedback resistor chosen value is $R_{feedback} = 1M\Omega$. Consequently, this amplifier has a current-to-voltage conversion factor of 10^6 (per example, an input current of 1uA should generate an output voltage of 1V).

A -10V reference voltage (V_{ref}) is applied to the positive input (V_+) of the op-amp to polarize the probe (connected to the negative input V_- , and considering an ideal op-amp with $V_+ = V_-$). The probe input current is simulated using a current source with a current value named "Iprobe".

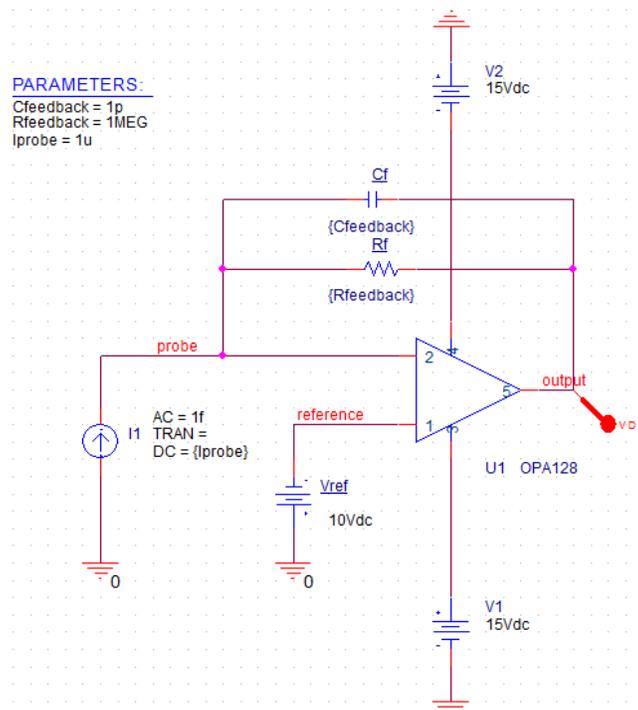


Figure 2. Trans-Impedance Amplifier circuit simulated

The power consumption of this amplifier is almost constant for different values of input current, and it is 13.7mW and 13.5mW for the negative and positive power supplies, respectively. This indicates that the TIA has a total consumption of approximately 27.2mW.

2.2. Transfer Function

Results of simulating different input currents are shown in Figure 3. These results indicate that the circuit is able to measure currents up to 4 μ A, where the operational amplifier output saturates.

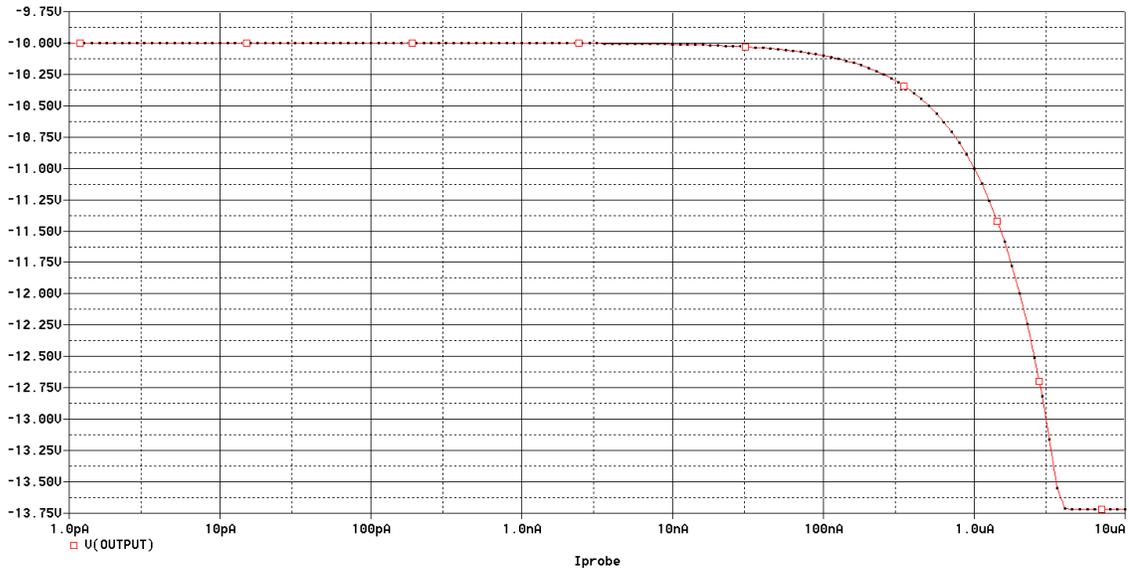


Figure 3. Simulation output obtained for different values of input probe current (I_{probe})

The TIA output signal has the -10V probe voltage added and it must be subtracted using an Instrumentation Amplifier (IA). The simulating signal corresponding to this subtraction is shown in Figure 4. This simulation shown that the maximum output voltage would be approximately 3.7V.

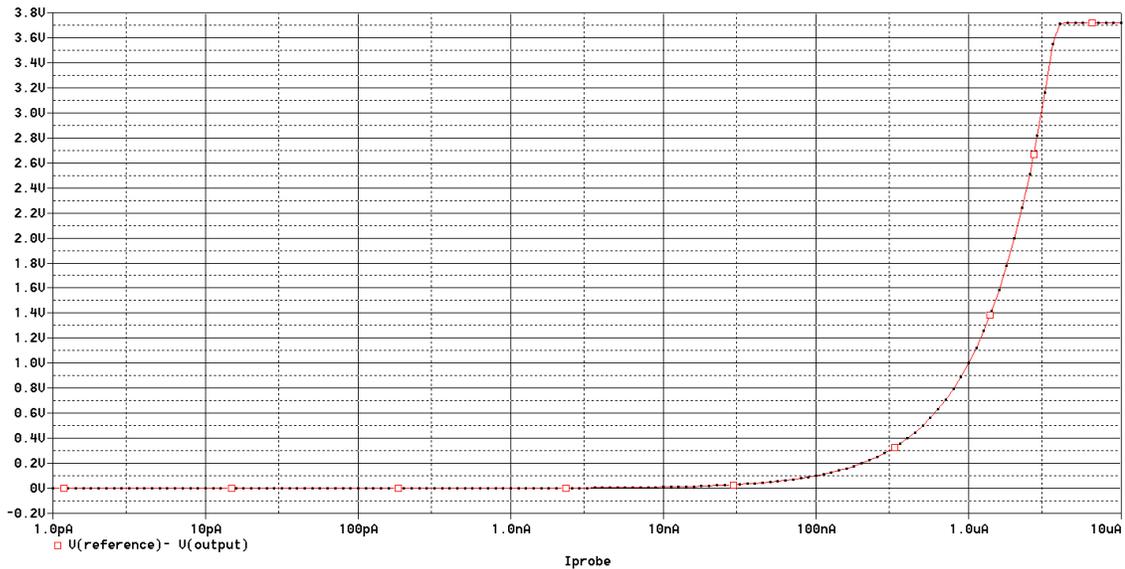


Figure 4. Simulation output subtracting the negative reference voltage

Also the output voltage has an offset of 7.0935 μ V, indicating that the minimum current that could be measured is 7.0935pA. Nevertheless, this offset could be compensated using a trimming resistance in the real OPA128 op-amp design.

2.3. Frequency Response

A common problem of TIA amplifiers is the effect known as “ringing” or “oscillations”. This is an enhancement of a specific frequency that leads to output stability problems. In simple words this means that the circuit starts oscillating at a predefined frequency. This is because a signal or noise at this frequency is consecutively amplified in the feedback loop until it obtains the maximum output level, or it gets damped (depending on the circuit frequency response). The undesired oscillation frequency depends on the op-amp frequency response, and the probe input capacitance (value unknown and dependent on the specific final hardware mounting). This undesired effect has been obtained during the mounting and testing of the current LP sensor circuit.

A common method to avoid this effect is applying a frequency compensation by using a feedback capacitor parallel to the feedback resistor ($C_{feedback}$ in our design). Nevertheless the addition of this capacitor changes the frequency response of the amplifier.

Figure 5 in its bottom section shows the frequency enhancement for very small capacitor values. A graphic of the cutoff frequency versus the capacitor value is shown in the middle section of this figure, and the maximum amplitude value it obtains in the upper section. From these graphics it is clear that the oscillating effect can be generated with a frequency of 427.5 kHz and it is compensated by increasing the feedback capacitor value. The value that provides the maximum bandwidth is approximately 0.625pF, with a cutoff frequency of 320 kHz. With 1pF and 10pF the cutoff frequency reduces to 182.5 kHz and 15.9 kHz, respectively.

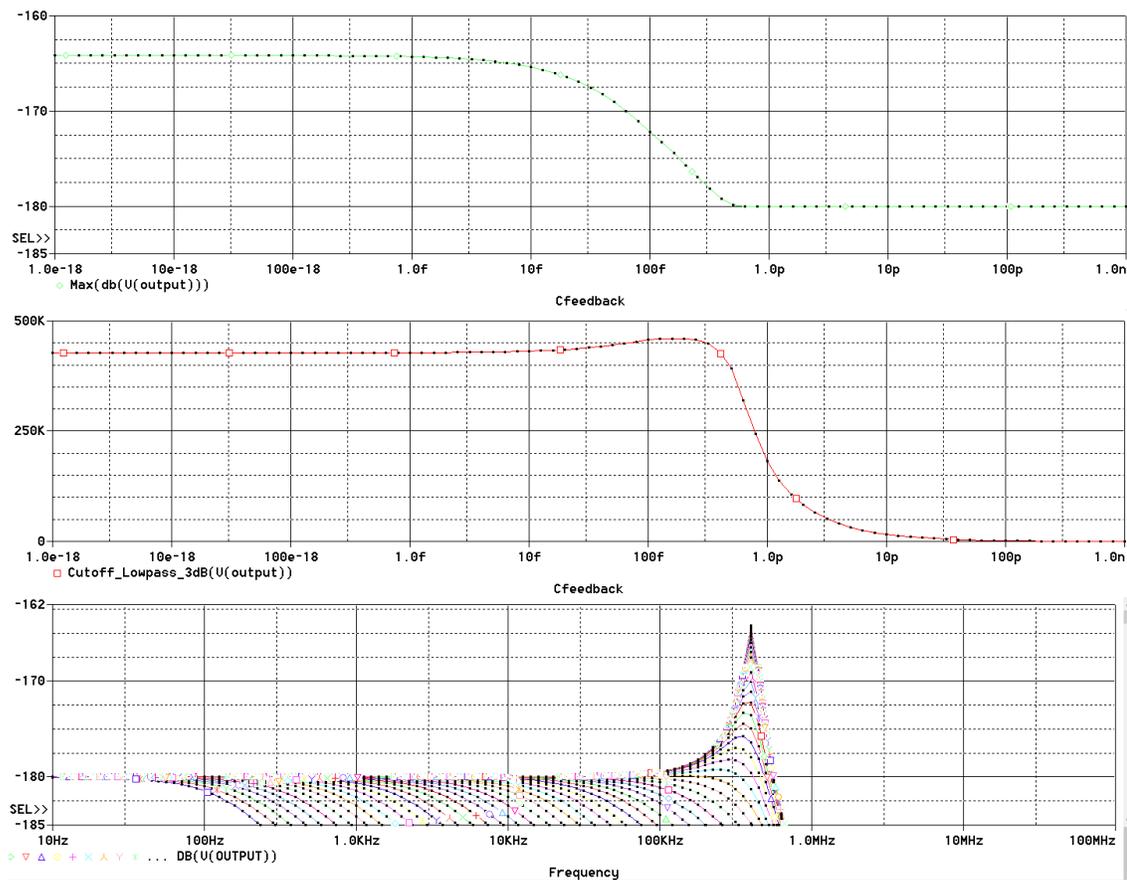


Figure 5. (Bottom) Frequency response, (Middle) Cutoff frequency, and (Up) Maximum amplitude for different feedback capacitor values

3. Complete Langmuir Probe Circuit

3.1. Design

The complete design is shown in Figure 6. The TIA amplifier with an op-amp OPA128 is located at the initial section, followed by the Instrumentation Amplifier (IA) with an AD8221 that has a gain of 2 (using a $R_g = 49.9\text{ k}\Omega$). An RC filter is used to delete high frequency signals on the Low Gain output, with a cutoff frequency of 16 kHz approximately. The High Gain output amplifies with a factor of 101 the Low Gain output with two op-amps OPA227.

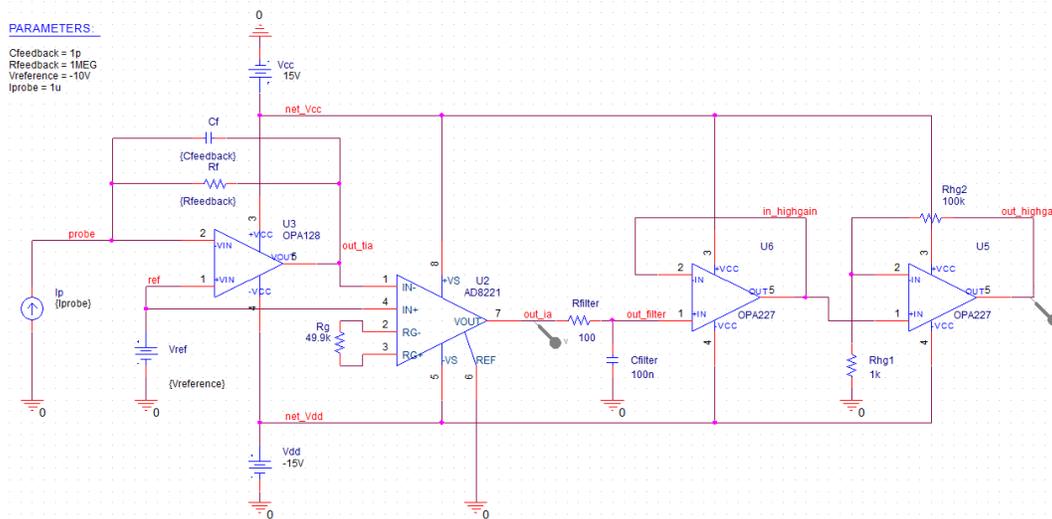


Figure 6. Complete circuit simulation design

3.2. Transfer Function

The simulated response of the circuit is shown in Figure 7. From this image it can be seen that the Low and High Gain output saturation are found at $3\ \mu\text{A}$ and 70 nA , respectively. These values are the maximum input currents that can be measured by each output channel.

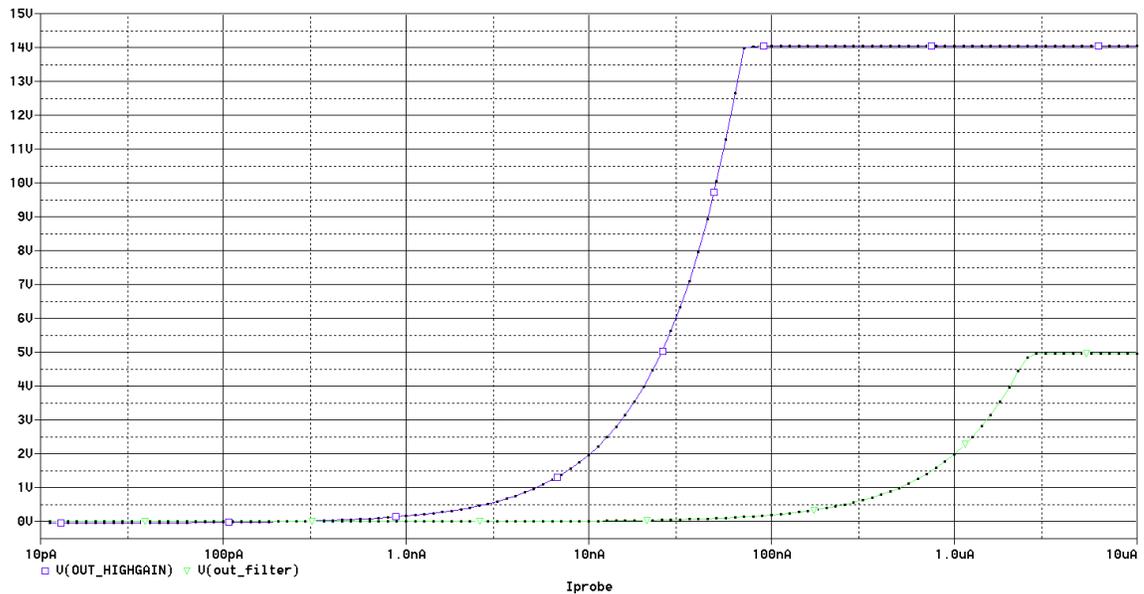


Figure 7. Low Gain and High Gain outputs for a sweep of input current

The output offset of the Low and High Gain have been calculated with a zero input current. These values are -370.43uV and -37.161mV for Low and High Gain outputs, respectively. The Low Gain value is 100.31 times smaller than the High Gain offset, which corresponds to the amplification introduced in the High Gain op-amp (101) plus an extra offset.

The Low Gain offset (-370.43uV) is much higher than the offset measured in the TIA simulation (of only 7.0935uV, although it should be multiplied by 2 because of the IA gain factor). Consequently, this means that this huge offset is related to the Instrumentation Amplifier used model AD8221. The Low Gain offset value obtained indicates that the minimum current that can be measured is approximately 185pA.

3.3. Frequency Response

Upper section of Figure 8 shows a graphic of the Instrumentation Amplifier frequency response depending on the feedback capacitance value, similar to the obtained in section 2.3. Alternatively, the middle section of this figure shows the frequency response of the Low Gain and High Gain outputs. In this section it is clear the effect of the Low Pass Filter, attenuating the high frequency components. The -3dB cutoff frequency of Low Gain and High Gain outputs is shown in the bottom section of this figure. In this image it can be seen that for small values of feedback capacitance the cutoff frequency is 17.086 kHz and 16.318 kHz for the Low and High Gain outputs, respectively. For values of feedback capacitance higher than 1pF the cutoff frequency is being reduced. Per example, for 10pF feedback capacitance the cutoff frequencies are 10.438 kHz and 10.592 kHz for the Low and High Gain outputs, respectively.

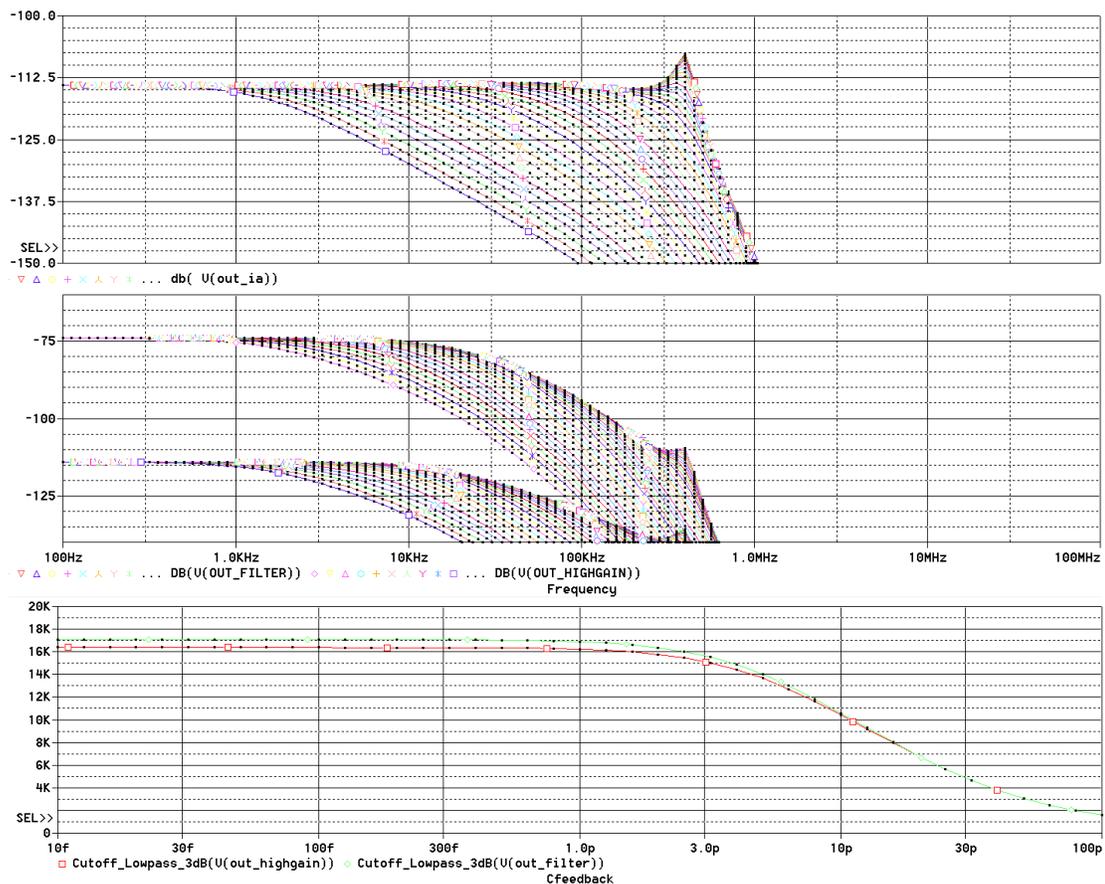


Figure 8. (Up) Instrumentation Amplifier frequency response, (Middle) Low Gain and High Gain frequency responses and (Bottom) Low Gain and High Gain cutoff frequencies for a sweep of feedback capacitance

3.4. Consumption

Figure 9 bottom section shows the power consumption drawn by the circuit at the positive and negative power supplies for different input currents. The upper section of this figure shows the total power consumption, adding both power supply consumptions. The maximum consumption value (approximately 320mW) is obtained when the High Gain output reaches saturation. Smaller input current values give a minimum power consumption of 297.7mW.

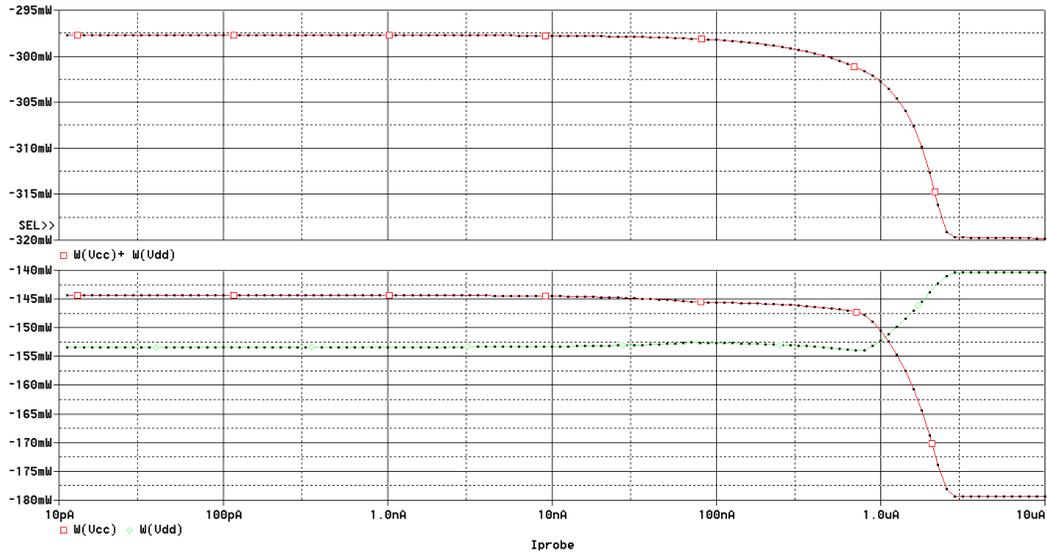


Figure 9. (Bottom) Positive and negative sources and (Top) total power consumptions for different input currents

3.5. Reference Noise

A simulation has been done adding a noisy signal to the -10V negative reference voltage, in order to estimate the effects of noise in this component. Bottom section of Figure 10 shows the addition of a 1 kHz sinusoid signal to the voltage reference with a variable amplitude. Upper section of this figure shows the amplitude (peak-to-peak value) of the Low Gain output for different values of reference voltage noise amplitude.

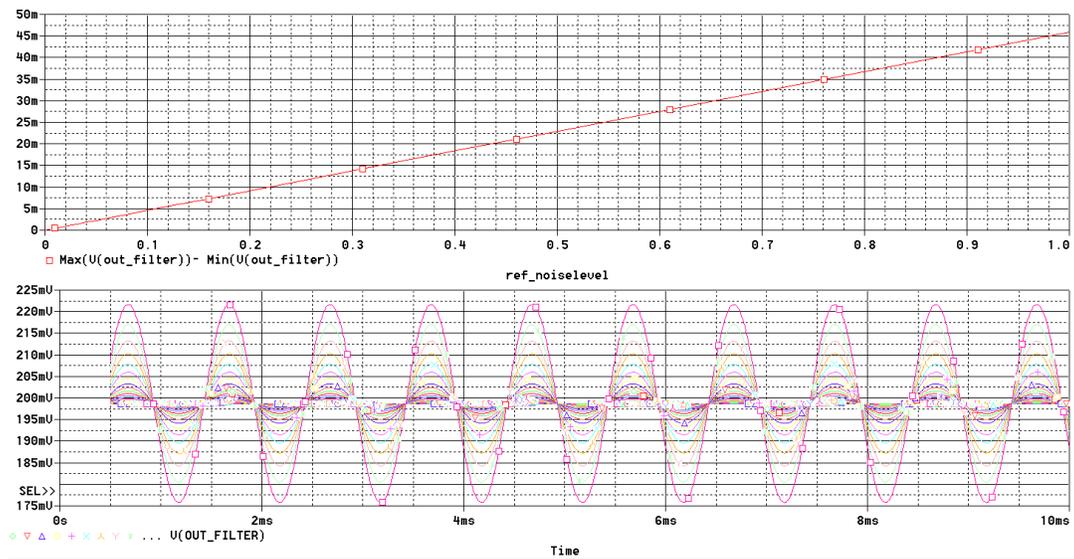


Figure 10. (Bottom) Temporal variability and (Top) peak-to-peak amplitude of Low Gain output for different amplitude values of noise added to the negative reference voltage

In previous figure it can be seen that the increase of noise in the reference voltage generates a linear increase of output noise amplitude. Nevertheless, note that for 1V reference noise addition (a 10% of the reference voltage value) only 46mV are obtained at the Low Gain output.

Further simulations have been done adding noise with different frequencies. Figure 11 shows the results of a 0.01V noise added (a 0.1% of the reference value). Results indicate that a noise with higher frequency has more relevance at the output, mainly because the op-amp circuitry is not able to compensate this input effect. Alternatively, when the noise frequency is smaller than 1kHz the output error is smaller than 0.5mV, which corresponds to 2.7nA of error. At frequencies smaller than 50Hz the output error becomes even smaller, with a value of 0.2mV which corresponds to 85pA of error.

It is relevant to highlight that in this simulation no Low Pass filter or capacitor has been placed to minimize fluctuations or noise in the voltage reference signal. In the real circuit a capacitor with 1uF is placed near the MCX output connector, reducing the possible input noise.

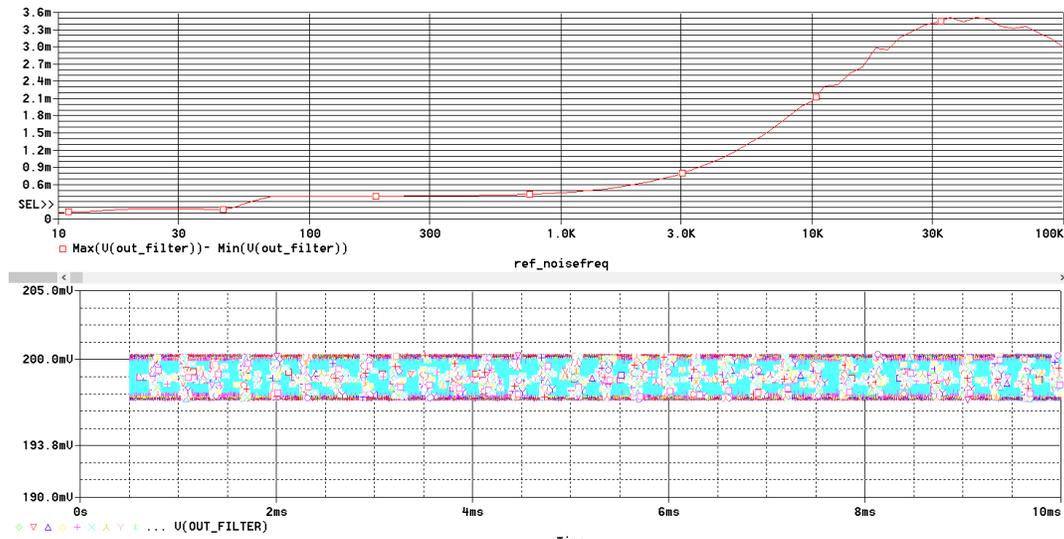


Figure 11. (Bottom) Temporal variability and (Top) peak-to-peak amplitude of Low Gain output for different values of noise frequency added to the negative reference voltage

4. Temperature Circuit

4.1. Design

The temperature circuit designed is based in a AD590 temperature sensor. This sensor provides a current output linear with $1\mu\text{A}/^\circ\text{K}$, and very small error and non-linearity. This current is converted into a voltage using a $1\text{k}\Omega$ resistor, and posteriorly amplified using an op-amp in a non-inverting configuration. The $1\text{mV}/^\circ\text{K}$ signal is amplified 11 times and also low-pass filtered to a frequency of approximately 16 kHz. This design is shown in Figure 12.

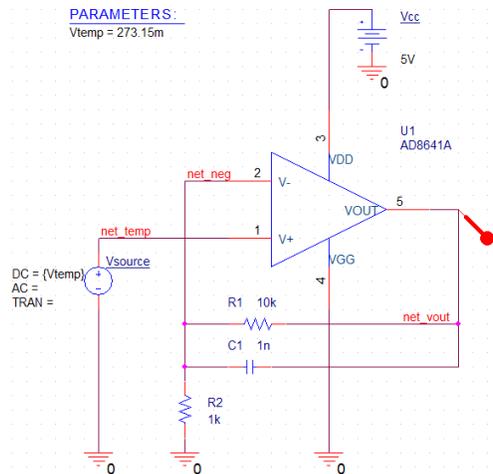


Figure 12. Temperature circuit schematic.

4.2. Transfer Function

The simulation of the output voltage for different input voltages is shown in Figure 13. From this image it is clear the amplification factor obtained and the small offset introduced. An input signal of 400mV obtains an output of 4.383V , and an input signal of 0V obtains 17.898mV . This indicates that the offset is 17.898mV which corresponds to an amplification of 1.0912755 .

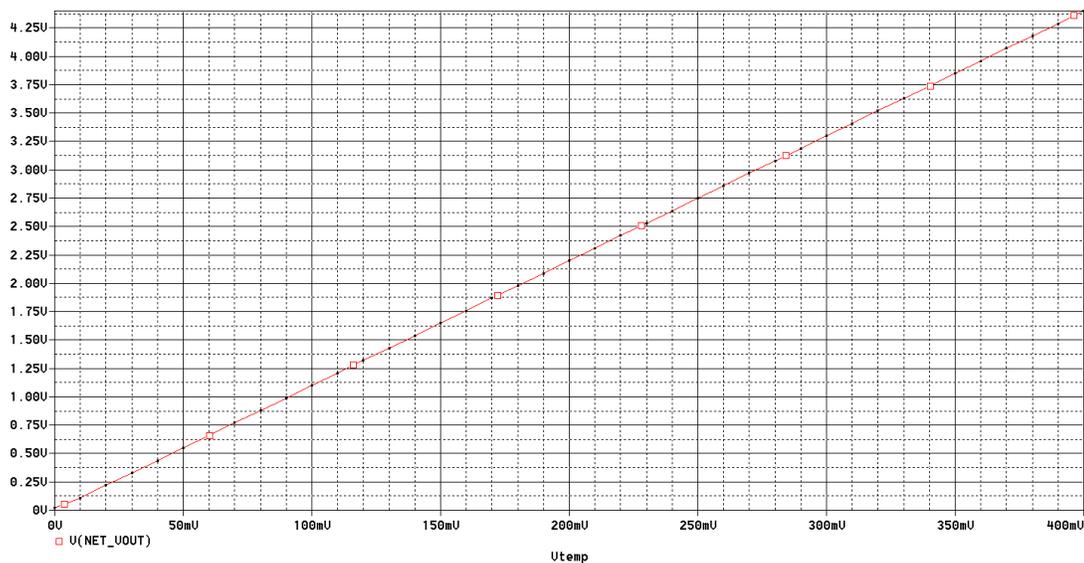


Figure 13. Output response for different input voltages.

4.3. Frequency Response

The theoretical frequency response of this circuit is shown in Figure 14, simulated using MATLAB. This theoretical simulation obtain a cutoff frequency of 15.92 kHz.

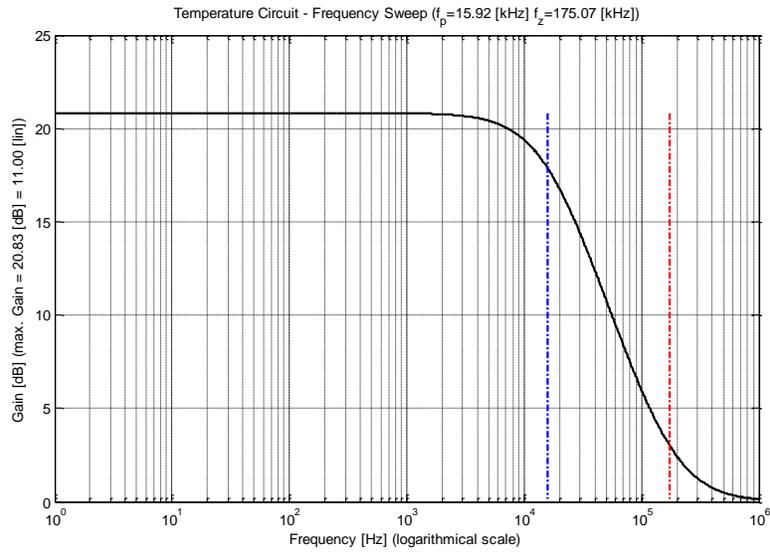


Figure 14. Theoretical frequency response simulated in MATLAB. Blue cursor indicates the -3dB cutoff frequency.

Alternatively, PSpice simulation shown in Figure 15 indicates that the cutoff frequency is 15.203 kHz. This is because of the more realistic model of op-amp used in this simulation.

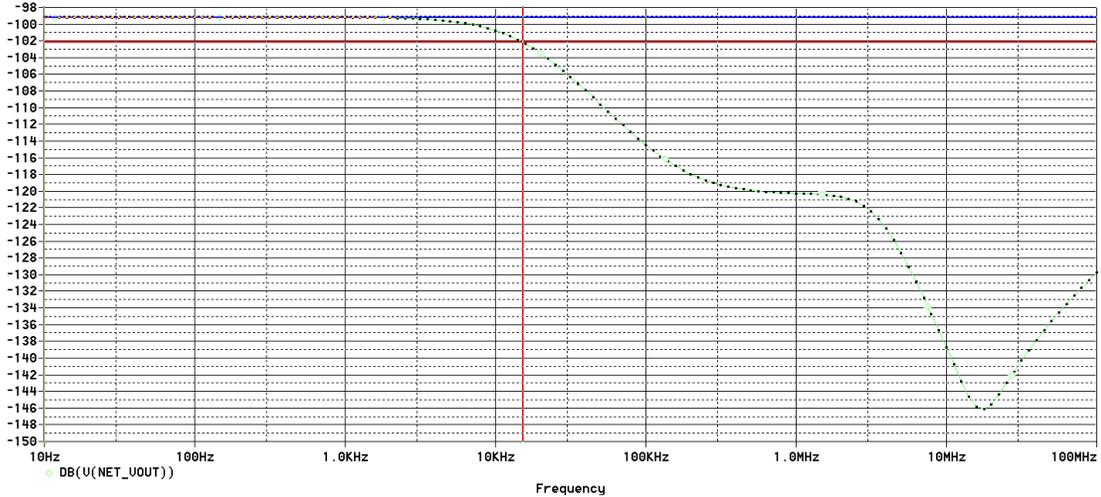
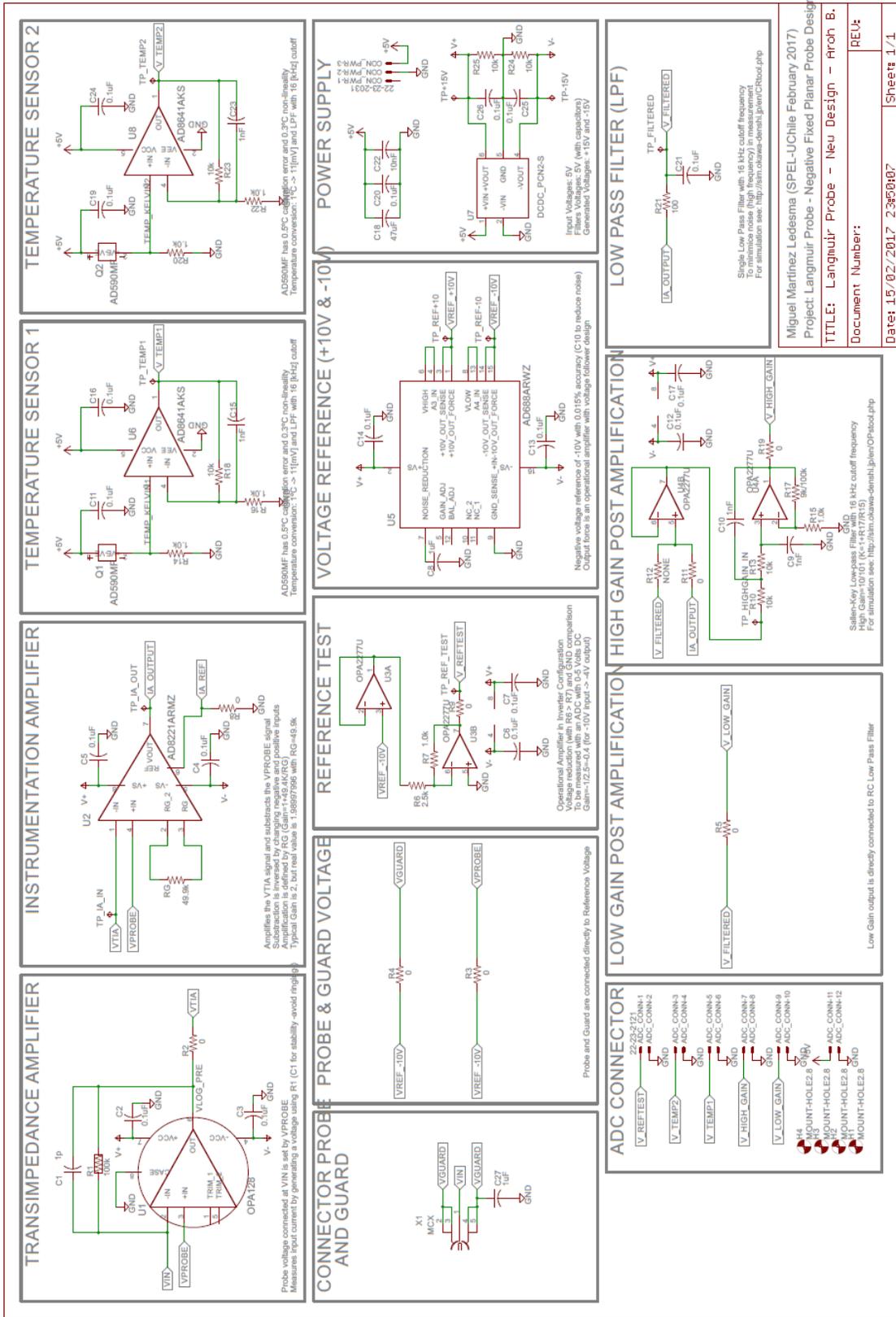


Figure 15. Frequency response. Red cursor indicates the -3dB cutoff frequency.

5. Annex - Schematic



A.3. Planar Langmuir Probe - Measurement Sheet.pdf

Planar Fixed Bias Langmuir Probe for Ram Measurement - Design Sheet

Probe Characteristics

Probe Dimensions (Rectangular)	Height (cm)	Width (cm)	2
Probe angle to Ram Direction	0 (deg)		

Probe Area

Probe Surface Area	0.0004 (m ²)
Probe Effective Projected Area	0.0004 (m ²)

e0	8.85E-12
kB	1.38E-23
qe	1.60E-19
amu	1.66E-27
me	9.11E-31

Ionspheric Characteristics

Ionspheric Ion Density	min NI (m ⁻³)	max NI (m ⁻³)		
Ionspheric Ion Temperature	9.00E+09	5.00E+11		
Ionspheric Composition	Ion Type	Ion Percentage	Ion Mass (AMU)	Ion Final Mass
	O+	85	16.0	14.563355
	NO+/O2+	2.5	30.5	
	He+	1.0	1.0	
	He+	2.5	4.0	
	Total Perc.	100		

Orbital Characteristics

Spacecraft Speed	min Vsc (m/s)	7924.36
	max Vsc (m/s)	7718.25

Circuit Characteristics

Transimpedance Amplification Factor	1.00E+06 (Current to Voltage Amplification)
Instrumentation Amplifier Factor	1.98977916 (Fixed Bias Compensation)
Low Gain Post Amplification	1 (Low Gain Amplification Output)
High Gain Post Amplification	101 (High Gain Amplification Output)

ADC Characteristics

Number of Bits	16 (bits)
Number of Levels	65536 (levels)
Minimum Input Voltage	0 (V)
Maximum Input Voltage	5 (V)

Input Thermal Noise

Reception Bandwidth	1.60E+04 (Hz)
Reception Temperature	100 (°C)

Localization Error Effects

Spacecraft speed variability	100 (m/s)
Spacecraft angle variability	10 (°)

Plasma Characteristics

min Plasma Debye Length	32.3318 (mm)	max Plasma Debye Length	4.3646 (mm)
Average Ion Velocity	852.61 (m/s)	$v_{Te} = \sqrt{\frac{2k_B T_e}{m_e}}$	$\lambda_{D0} \equiv \left(\frac{60k_B T_e}{n_e q_e^2} \right)^{1/2}$
Average Electron Velocity	##### (m/s)		

Thermal Collected Current

min Thermal Ion Saturation Current	2.459E-10 (A)	max Thermal Ion Saturation Current	1.366E-08 (A)
			$I_{Thermal,s} = \frac{1}{2} n_e q_e A v_{Te}$

Ram Collected Current

min Speed & min Density Current	4.282E-09 (A)	min Speed & max Density Current	2.379E-07 (A)
max Speed & min Density Current	4.452E-09 (A)	max Speed & max Density Current	2.473E-07 (A)
min Current Difference (diff. Speed)	3.958 (%)	max Current Difference (diff. Speed)	3.958 (%)
			$I_{ram,s} = n_e q_e A v_{Ts}$

Difference between Ram and Thermal Currents	min Speed	5.742 (%)
	max Speed	5.528 (%)

Minimum Density Output Voltages

Transimpedance Amplifier	4.28E-08 (V)
Instrumentation Amplifier	8.52E-08 (V)
Low Gain Amplifier	0.000522 (V)
High Gain Amplifier	0.860561 (V)

Maximum Density Output Voltages

Transimpedance Amplifier	2.47E-01 (V)
Instrumentation Amplifier	4.92E-01 (V)
Low Gain Amplifier	0.4492162 (V)
High Gain Amplifier	49.2789343 (V)

Low Gain Post Amplification ADC Step Counts

ADC Voltage Step	7.63E-05 (V)	ADC Voltage Step	7.63E-05 (V)
Low Gain Amplification Step	7.63E-05 (V)	High Gain Amplification Step	7.5539E-07 (V)
Instrumentation Amplifier Step	3.89E-05 (V)	Instrumentation Amplifier Step	3.80E-07 (V)
Transimpedance Amplifier Step	3.89E-04 (A)	Transimpedance Amplifier Step	3.80E-08 (A)
Density Step (min Speed)	8.06E-07 (m ⁻³)	Density Step (min Speed)	7.99E-06 (m ⁻³)
Max Density Value (max Voltage)	5.28E+12 (m ⁻³)	Max Density Value (max Voltage)	5.23E+10 (m ⁻³)
Max Density Design Objective Error	956.1 (%)	Max Density Design Objective Error	-893.5 (%)
Min Density Voltage	0.000522 (V)	Min Density Voltage	0.86060592 (V)
Min Density ADC Value	112 (Steps)	Min Density ADC Value	11281 (Steps)
Min Density ADC Range Used	0.2 (%)	Min Density ADC Range Used	17.2 (%)

Thermal Noise Levels

Johnson Noise on Input TIA Resistor	1.82E-11 (A)	$i_n = \sqrt{\frac{4k_B T_e}{R_{in}}}$	High Gain Density Noise	3.82E+07 (m ⁻³)
Low Gain Density Noise	3.82E+07 (m ⁻³)		Johnson of High Gain ADC Resolution	4783.6718 (m ⁻³)
Johnson of Low Gain ADC Resolution	47.3631 (m ⁻³)			48 (Steps)

min Density Speed Variation

Input Current Error	5.77E-11 (A)
Density Error	1.21E+08 (m ⁻³)
Density Error Percentage	1.34692 (%)
Low Gain ADC Steps	1.5 (Steps)
High Gain ADC Steps	151.9 (Steps)

max Density Speed Variation

Input Current Error	3.20E-09 (A)
Density Error	6.48E+09 (m ⁻³)
Density Error Percentage	1.29563 (%)
Low Gain ADC Steps	1.5 (Steps)
High Gain ADC Steps	8120.1 (Steps)

min Density Angle Variation

Input Current Error	-6.51E-11 (A)
Density Error	-1.37E+08 (m ⁻³)
Density Error Percentage	-1.51922 (%)
Low Gain ADC Steps	-1.7 (Steps)
High Gain ADC Steps	-171.4 (Steps)

max Density Angle Variation

Input Current Error	-3.76E-09 (A)
Density Error	-7.60E+09 (m ⁻³)
Density Error Percentage	-1.51922 (%)
Low Gain ADC Steps	-94.3 (Steps)
High Gain ADC Steps	9521.4 (Steps)

Anexos B

B.1. Code: 'Main.c'

```
/*                      SUCHAI
*                      NANOSATELLITE FLIGHT SOFTWARE
*
*    Copyright 2018, Carlos Gonzalez Cortes, carlgonz@uchile.cl
*    Copyright 2018, Camilo Rojas Milla, camrojas@uchile.cl
*    Copyright 2018, Tomas Opazo Toro, tomas.opazo.t@gmail.com
*    Copyright 2018, Matias Ramirez Martinez, nicoram.mt@gmail.com
*    Copyright 2018, Tamara Gutierrez Rojo TGR_93@hotmail.com
*    Copyright 2018, Ignacio Ibanez Aliaga, ignacio.ibanez@usach.cl
*
* This program is free software: you can redistribute it and/or modify
* it under the terms of the GNU General Public License as published by
* the Free Software Foundation, either version 3 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program. If not, see <http://www.gnu.org/licenses/>.
*/

#include "main.h"

const char *tag = "main";
```

```

#ifdef ESP32
void app_main()
#else
int main(void)
#endif
{

    /* On reset */
    on_reset();

    printf("\n\n----0----- Inicio del main (FLIGHT SOFTWARE START) -----\n\n");

    /* Init software subsystems */
    log_init();          // Logging system
    cmd_repo_init();    // Command repository initialization
    dat_repo_init();    // Update status repository
    printf("\n\n----0.1----- Inicio del main (FLIGHT SOFTWARE START) -----\n\n");

    /* Initializing shared Queues */
    dispatcher_queue = osQueueCreate(25, sizeof(cmd_t *));
    if(dispatcher_queue == 0)
        LOGE(tag, "Error creating dispatcher queue");
    executer_stat_queue = osQueueCreate(1, sizeof(int));
    if(executer_stat_queue == 0)
        LOGE(tag, "Error creating executer stat queue");
    executer_cmd_queue = osQueueCreate(1, sizeof(cmd_t *));
    if(executer_cmd_queue == 0)
        LOGE(tag, "Error creating executer cmd queue");
    printf("\n\n----0.2----- Inicio del main (FLIGHT SOFTWARE START) -----\n\n");
    int n_threads = 4; // en el main original de la radiosonda es 3
    os_thread threads_id[n_threads];

    LOGI(tag, "Creating basic tasks...");
    /* Crating system task (the others are created inside taskInit) */
    int t_inv_ok = osCreateTask(taskDispatcher, "invoker", //por espacio se cortó
    // SCH_TASK_DIS_STACK, NULL, 3, &threads_id[1]);
    int t_exe_ok = osCreateTask(taskExecuter, "receiver", //por espacio se cortó
    SCH_TASK_EXE_STACK, NULL, 4, &threads_id[2]);
    int t_wdt_ok = osCreateTask(taskWatchdog, "watchdog", //por espacio se cortó
    SCH_TASK_WDT_STACK, NULL, 2, &threads_id[0]);
    int t_ini_ok = osCreateTask(taskInit, "init", //por espacio se cortó
    SCH_TASK_INI_STACK, NULL, 3, &threads_id[3]);
    printf("\n\n----0.3----- Inicio del main (FLIGHT SOFTWARE START) -----\n\n");
    /* Check if the task were created */
    if(t_inv_ok != 0) LOGE(tag, "Task invoker not created!");
    if(t_exe_ok != 0) LOGE(tag, "Task receiver not created!");
    if(t_wdt_ok != 0) LOGE(tag, "Task watchdog not created!");
}

```

```

    if(t_ini_ok != 0) LOGE(tag, "Task init not created!");
    printf("\n\n----0.4----- Inicio del main (FLIGHT SOFTWARE START) ----- \n\n");
    //config_LP();
    //while(1){
    // measure_LP();
    // delay_ms(38); // la impresion de datos tendria un periodo de 50ms y una
                    // frecuencia de 20Hz
    //}
    config_LP();
    while(1){
        measure_LP();
        //delay_ms(3);
        printf("\n\n----NO ESTA IMPRIMIENDO----- \n\n");
    }
}

#ifdef ESP32
    /* Start the scheduler. Should never return */
    osScheduler(threads_id, n_threads); //(Sistema operativo)
    return 0;
#endif

}

#ifdef FREERTOS
#ifdef NANOMIND
/**
 * Task idle handle function. Performs operations inside the idle task
 * configUSE_IDLE_HOOK must be set to 1
 */
void vApplicationIdleHook(void)
{
    //Add hook code here
}

/**
 * Task idle handle function. Performs operations inside the idle task
 * configUSE_TICK_HOOK must be set to 1
 */
void vApplicationTickHook(void)
{
#ifdef AVR32
    LED_Toggle(LED0);
#endif
}
}

```

```

/**
 * Stack overflow handle function.
 * configCHECK_FOR_STACK_OVERFLOW must be set to 1 or 2
 *
 * @param pxTask Task handle
 * @param pcTaskName Task name
 */
void vApplicationStackOverflowHook(xTaskHandle* pxTask, signed char* pcTaskName)
{
    printf("[ERROR] [-1] [%s] Stack overflow!", (char *)pcTaskName);

    /* Stack overflow handle */
    while(1);
}
#endif
#endif

```

B.2. Code: 'langmuirProbe.h'

```

//
// Created by kaminari on 10-07-18.
//

#ifndef _SUCHAI_FLIGHT_SOFTWARE_LANGMUIR_H_
#define _SUCHAI_FLIGHT_SOFTWARE_LANGMUIR_H_

#include "config.h"
#include "repoCommand.h"
#include "init.h"
// #include "main.h"
// #include "asf.h"
#include "perifericos.h"
#include "ADS1298.h"
#include <spi.h>
#include <time.h>
#include <stdio.h>
// #include "conf_spi_master.h"

#include "sd_mmc_spi.h"
// #include <string.h> //SD diego
// #include "setup.h" //SD diego
// #include <ff.h> //SD diego
// #include <integer.h> //SD diego

```

```

/**interrupcion**----->0.0
#include <stdint.h>
#include <stdbool.h>

#include "sysclk.h"
#include "board.h"
#include "print_funcs.h"
#include "tc.h"
#include "gpio.h"
#if defined (__GNUC__)
# include "intc.h"
#endif
#include "conf_example.h"
#include "conf_clock.h"
#include "conf_board.h"

#if !defined(EXAMPLE_TC) || !defined(EXAMPLE_TC_IRQ)
# error The TC preprocessor configuration to use in this example is missing.
#endif
/**interrupcion**-----<0.0

//=====lll
#define conf_pin_ADC_CS          gpio_enable_gpio_pin(AVR32_PIN_PA07)
#define conf_pin_ADC_reset      gpio_enable_gpio_pin(AVR32_PIN_PA31)
#define conf_pin_ADC_convst     gpio_enable_gpio_pin(AVR32_PIN_PA29)

// **configure as input**
#define conf_pin_ADC_BUSY      gpio_enable_gpio_pin(AVR32_PIN_PA30)

// **put as HIGH**
#define set_pin_ADC_CS_HIGH    gpio_set_gpio_pin(AVR32_PIN_PA07)
#define set_pin_ADC_reset_HIGH gpio_set_gpio_pin(AVR32_PIN_PA31)
#define set_pin_ADC_convst_HIGH gpio_set_gpio_pin(AVR32_PIN_PA29)

// **put as LOW**
#define set_pin_ADC_CS_LOW     gpio_clr_gpio_pin(AVR32_PIN_PA07)
#define set_pin_ADC_reset_LOW  gpio_clr_gpio_pin(AVR32_PIN_PA31)
#define set_pin_ADC_convst_LOW gpio_clr_gpio_pin(AVR32_PIN_PA29)

// **read pin**
#define read_pin_ADC_BUSY      gpio_get_pin_value(AVR32_PIN_PA30)

/**
 * Register on board computer related (OBC) commands

```

```

*/
/*void cmd_subsys_init(void);

int get_gps_data(char *fmt, char *params, int nparams);
int get_dpl_data(char *fmt, char *params, int nparams);
int get_prs_data(char *fmt, char *params, int nparams);
int open_dpl_la(char *fmt, char *params, int nparams);
int close_dpl_la(char *fmt, char *params, int nparams);
int open_dpl_sm(char *fmt, char *params, int nparams);
int close_dpl_sm(char *fmt, char *params, int nparams);
int send_iridium_data(char *fmt, char *params, int nparams);
int send_iridium_msg1(char *fmt, char *params, int nparams);
int send_iridium_msg2(char *fmt, char *params, int nparams);*/
// void spi_init_master (void);
// void spi_init_slave (void);

void spi_init_module(void);
void resetADC(void);
void startConversion(void);
void startAcquisition(void);
void readADC(unsigned char show, unsigned int bravo);
void writeHeader(void);
void logData(void);
unsigned char spi_tranceiver (unsigned char data);
void config_LP(void);
void measure_LP(void);

#endif // _SUCHAI_FLIGHT_SOFTWARE_LANGMUIR_H_

```

B.3. Code: 'langmuirProbe.c'

```

/*
*****
*****
*****
*****
*****
*****
*****SOFTWARE DEL PAYLOAD LANGMUIR PROBE*****
*****
*****
*****
*****
*****

```

```

*****
*/

//
// Created by kaminari on 10-07-18.
//
#include "langmuirProbe.h"
static const char* tag = "langmuirProbe";

uint16_t arreglo_1 [1299]; //2000 es el ideal
uint16_t arreglo_2 [1299]; //2000 es el ideal
//int* puntero = arreglo;
//int32_t diff;
const uint8_t ANALOG_COUNT = 6;
// unsigned long mil = 1000;
void sysclk_init (void); // antes: void sysclk_init (void);
// Buffer to send data to SPI slave
uint16_t txdata;
//Buffer to receive data from SPI slave
uint16_t rxdata;

clock_t seconds;

//***interrupción***----->0.1
// Flag to update the timer value
volatile static bool update_timer = true;
volatile static bool update_timer_available = true; // *****
volatile static bool se_puede_imprimir = false; // *****
//volatile static bool se_puede_copiar = false; // *****MIGUEL
volatile static bool escribiendo = true; // *****

// Variable to contain the time ticks occurred
volatile static uint32_t tc_tick_ms = 0;
volatile static uint32_t tc_tick_s = 0;
//***interrupción***-----<0.1

//***interrupción***----->1

#if defined (__GNUC__)
__attribute__((__interrupt__))
#elif defined (__ICCAVR32__)
#pragma handler = EXAMPLE_TC_IRQ_GROUP, 1
__interrupt
#endif

static void tc_irq(void) { // TC interrupt

```

```

// Clear the interrupt flag. This is a side effect of reading the TC SR.
tc_read_sr(EXAMPLE_TC, EXAMPLE_TC_CHANNEL);
// Increment the ms seconds counter
tc_tick_ms++;

//if (true) { //MIGUEL
//tc_tick_ms>=100){
//tc_tick_ms>=1000){ //<-----original
tc_tick_ms = 0;
tc_tick_s++;
// create a semaphore
//if (update_timer_available == true) { //MIGUEL
//    update_timer_available = false; //MIGUEL

//-----
//--Critical Section---(START)
//-----
//update_timer = true; // **** original MIGUEL

startConversion();
if (tc_tick_s <= 1299) { //1299
    for (uint8_t i = 0; i < ANALOG_COUNT; i++) {
        readADC(i, tc_tick_s - 1); //start
    }
} else {
    //se_puede_copiar = true; MIGUEL
    //} //MIGUEL

    if (se_puede_imprimir == false) { //MIGUEL
        //se_puede_imprimir == false && se_puede_copiar == true) {
        for (uint16_t k = 0; k < 1299; k++) {
            arreglo_2[k] = arreglo_1[k];
        }
        //en este momento el segundo vector puede ser impreso

        se_puede_imprimir = true;
        //se_puede_copiar = false; //MIGUEL
        //tc_tick_s = 0;
    }
    tc_tick_s = 0; //MIGUEL
} //MIGUEL
//-----
//--Critical Section---(END)
//-----
//update_timer_available = true; MIGUEL
//} //MIGUEL
//} //MIGUEL

```

```

    // Toggle a GPIO pin (this pin is used as a regular GPIO pin).
    LED_Toggle(LED0);
}

static void tc_init(volatile avr32_tc_t *tc) {
    // Options for waveform generation.
    static const tc_waveform_opt_t waveform_opt = {
        // Channel selection.
        .channel = EXAMPLE_TC_CHANNEL,
        // Software trigger effect on TIOB.
        .bswtrg = TC_EVT_EFFECT_NOOP,
        // External event effect on TIOB.
        .beevt = TC_EVT_EFFECT_NOOP,
        // RC compare effect on TIOB.
        .bcpc = TC_EVT_EFFECT_NOOP,
        // RB compare effect on TIOB.
        .bcpb = TC_EVT_EFFECT_NOOP,
        // Software trigger effect on TIOA.
        .aswtrg = TC_EVT_EFFECT_NOOP,
        // External event effect on TIOA.
        .aeevt = TC_EVT_EFFECT_NOOP,
        // RC compare effect on TIOA.
        .acpc = TC_EVT_EFFECT_NOOP,
        /*
         * RA compare effect on TIOA.
         * (other possibilities are none, set and clear).
         */
        .acpa = TC_EVT_EFFECT_NOOP,
        /*
         * Waveform selection: Up mode with automatic trigger(reset)
         * on RC compare.
         */
        .wavscl = TC_WAVEFORM_SEL_UP_MODE_RC_TRIGGER,
        // External event trigger enable.
        .enetrg = false,
        // External event selection.
        .eevt = 0,
        // External event edge selection.
        .eevtedg = TC_SEL_NO_EDGE,
        // Counter disable when RC compare.
        .cpcdis = false,
        // Counter clock stopped with RC compare.
        .cpcstop = false,
        // Burst signal selection.
        .burst = false,
        // Clock inversion.

```

```

        .clki      = false,
        // Internal source clock 3, connected to fPBA / 8.
        .tcclks   = TC_CLOCK_SOURCE_TC3 // muy lento c:
        // TC_CLOCK_SOURCE_TC4 con este aumento el periodo (mas lento?)
        // TC_CLOCK_SOURCE_TC3 original
};

// Options for enabling TC interrupts
static const tc_interrupt_t tc_interrupt = {
    .etrgs = 0,
    .ldrbs = 0,
    .ldras = 0,
    .cpcs  = 1, // Enable interrupt on RC compare alone
    .cpbs  = 0,
    .cpas  = 0,
    .lovrs = 0,
    .covfs = 0
};
// Initialize the timer/counter.
tc_init_waveform(tc, &waveform_opt); // 1_prueba

/*
 * Set the compare triggers.
 * We configure it to count every 1 milliseconds.
 * We want:  $(1 / (fPBA / 8)) * RC = 1 \text{ ms}$ , hence  $RC = (fPBA / 8) / 1000$ 
 * to get an interrupt every 10 ms.
 */
tc_write_rc(tc, EXAMPLE_TC_CHANNEL, (sysclk_get_pba_hz() / 8 / 7200));
// F = 14.42k[Hz]
// (sysclk_get_pba_hz() / 8 / 7500)); // F = 15k[Hz]
// (sysclk_get_pba_hz() / 8 / 3100));
// maximo baudrate funcional F = 6,2k[Hz]
// (sysclk_get_pba_hz() / 8 / 31000));
// este alcanza una F = 31,25k[Hz] (m.s. no funciona)
// (sysclk_get_pba_hz() / 8 / 34000));
// sysclk_get_pba_hz() / 8 / 925));
// el mas rapido que funciona: F = 1,85k[Hz]
// (sysclk_get_pba_hz() / 8 / 100000));
// ((FOSCO / 8) * 10)); //<-----
// ((sysclk_get_pba_hz() / 8)*100));
// (sysclk_get_pba_hz() / 8 / 1000)); original
// configure the timer interrupt
tc_configure_interrupts(tc, EXAMPLE_TC_CHANNEL, &tc_interrupt);
// Start the timer/counter.
tc_start(tc, EXAMPLE_TC_CHANNEL);
}
//***interrupción***-----<1

```

```

/**interrupción***----->1.5
volatile avr32_tc_t *tc = EXAMPLE_TC;
/**interrupción***-----<1.5

void resetADC(void){
    set_pin_ADC_convst_HIGH;
    set_pin_ADC_reset_HIGH;
    delay_ms(1);
    set_pin_ADC_reset_LOW;
    delay_ms(1);
    set_pin_ADC_convst_LOW;
}

void writeHeader(void){
    printf("\nCada lectura (de 16 bits) se          //por espacio se cortó
// entrega como 2 bytes separados por '|'|.\n");
    printf("Sampling frequency = 83,3 [Hz].\n");
    printf("SAMPLE, HIGH_GAIN, LOW_GAIN,          //por espacio se cortó
// TEMP1, TEMP2, V5, V6 (no se utiliza).\n");
    printf("\nEXAMPLE_TC_IRQ_PRIORITY:%u\n",EXAMPLE_TC_IRQ_PRIORITY);
    printf("DATA_OUT_SPI_BAUDRATE = %u\n", DATA_OUT_SPI_BAUDRATE);

}

void config_LP(void) {
    conf_pin_ADC_CS;          // Output
    conf_pin_ADC_reset;
    conf_pin_ADC_convst;
    conf_pin_ADC_BUSY;       // Input

    set_pin_ADC_CS_HIGH; //<-----prueba
    set_pin_ADC_reset_LOW; //<-----prueba
    set_pin_ADC_convst_HIGH; //<-----prueba

    spi_init(); // originalmente spi_init(),
// funcion que inicializa la comunicacion SPI

    writeHeader();
    resetADC();

    /**interrupción***----->2
/**
 * \note the call to sysclk_init() will disable all non-vital
 * peripheral clocks, except for the peripheral clocks explicitly
 * enabled in conf_clock.h.
 */
//sysclk_init(); // *****

```

```

//printf("\n\n----0.41----- Inicio del main (FLIGHT SOFTWARE START) -----\n\n");
// Enable the clock to the selected example Timer/counter peripheral module.
    sysclk_enable_peripheral_clock(EXAMPLE_TC);
//printf("\n\n----0.42----- Inicio del main (FLIGHT SOFTWARE START) -----\n\n");
// Initialize the USART module for trace messages
//init_dbg_rs232(sysclk_get_pba_hz()); // *****
//ESTA FUNCION CAMBIA EL BAUDRATE DE TRANSFERENCIA
//printf("\n\n----0.43----- Inicio del main (FLIGHT SOFTWARE START) -----\n\n");
// Disable the interrupts
    cpu_irq_disable();
//printf("\n\n----0.5----- Inicio del main (FLIGHT SOFTWARE START) -----\n\n");
// Initialize interrupt vectors.
//INTC_init_interrupts();
// esta funcion detiene el monitor serial >:C
// Register the RTC interrupt handler to the interrupt controller.
    INTC_register_interrupt(&tc_irq, EXAMPLE_TC_IRQ, EXAMPLE_TC_IRQ_PRIORITY);
//printf("EXAMPLE_TC_IRQ_PRIORITY: %u\n", EXAMPLE_TC_IRQ_PRIORITY);
//EXAMPLE_TC_IRQ_PRIORITY
//printf("EXAMPLE_TC_IRQ_GROUP: %u", EXAMPLE_TC_IRQ_GROUP);
//EXAMPLE_TC_IRQ_PRIORITY
//printf("EXAMPLE_TC_IRQ: %u\n", EXAMPLE_TC_IRQ);
//EXAMPLE_TC_IRQ_PRIORITY
//printf("EXAMPLE_TC_CHANNEL: %u\n", EXAMPLE_TC_CHANNEL);
//EXAMPLE_TC_IRQ_PRIORITY
//printf("EXAMPLE_TC: %u\n", EXAMPLE_TC);
//EXAMPLE_TC_IRQ_PRIORITY

//printf("\n\n----0.6----- Inicio del main (FLIGHT SOFTWARE START) -----\n\n");
// Enable the interrupts
    cpu_irq_enable();
// Initialize the timer module
    tc_init(tc);
//***interrupción***-----<2

}
void startConversion(void){
    set_pin_ADC_convst_LOW;
    set_pin_ADC_convst_HIGH;
    while(read_pin_ADC_BUSY==1) {}
}
void startAcquisition(void){
    set_pin_ADC_convst_HIGH;
    set_pin_ADC_convst_LOW;
}
void readADC(unsigned char show, unsigned int bravo){
    uint16_t highbyte;
    uint16_t lowbyte;

```

```

uint8_t txdata = 0x00;
spi_selectChip(DATA_OUT_SPI, DATA_OUT_SPI_NPCS);
while (!spi_is_tx_empty(DATA_OUT_SPI)) {}
spi_put(DATA_OUT_SPI, txdata);
while (!spi_is_tx_empty(DATA_OUT_SPI)) {}
highbyte = spi_get(DATA_OUT_SPI);
spi_put(DATA_OUT_SPI, txdata);
while (!spi_is_tx_empty(DATA_OUT_SPI)) {}
lowbyte = spi_get(DATA_OUT_SPI);
spi_unselectChip(DATA_OUT_SPI, DATA_OUT_SPI_NPCS);
if (show == 4) { //if (show == 1) {
    arreglo_1[bravo] = ((uint16_t)(0x00FF & highbyte) << 8) | (0x00FF & lowbyte);
}
}
/*
void logData(unsigned char kak){
    startConversion();
    for (uint8_t i = 0; i < ANALOG_COUNT; i++) {
        readADC(i, kak); //readADC(1);
    }
}
*/
void measure_LP(void) {

    //if (update_timer_available == true) { //MIGUEL
    //    printf("\nwhile_1_if_1\n");
    //update_timer_available = false; //MIGUEL

    //-----
    //--Critical Section--(Start)
    //-----

    //if (update_timer) { //MIGUEL
    //printf("\nmeasure_LP_if_(u.t.)\n");
    //printf("-----\n");
    //printf("1_tc_tick_s: %u\n",tc_tick_s);
    //update_timer = false; //MIGUEL

    if(se_puede_imprimir == true){ //MIGUEL
        //se_puede_imprimir == true && se_puede_copiar == false){

        printf("2_tc_tick_s: %u\n",tc_tick_s);
        for (int loop = 0; loop < 1299; loop++){ //1299
            //printf("tc_tick_ms: %u \n", tc_tick_ms);
            printf(" arreglo_2[%u]: ", loop);
            printf("%u \n", arreglo_2[loop]);
        }
    }
}

```

```

        se_puede_imprimir = false;
        //tc_tick_s = 0;
    }
    //} //MIGUEL
    //-----
    //--Critical Section--(End)
    //-----

    //update_timer_available = true; //MIGUEL
    //} //MIGUEL
}

```

B.4. Code: 'FIT_load_and_graf_all_test_freq_inter.m'

```

clc; clear;

%% Configuration: Se agrega la carpeta en donde se encuentran los datos
oldpath = path;
roota = ('/Users/matiassandoval/Documents/Universidad-ex_pendrive/Memoria');
rootb = ('/Calibraciones/calibracion_adc_avr32_uc3_frecuencia_interruccion');
rootc = ('/datos_calibra_interruccion/');
path(strcat(roota,strcat(rootb,rootc)),oldpath)
%% las variables de entrada de la funcion son:
% el nombre de la prueba y la frecuencia del ADC (esta ultima medida con el
% ADC_BUSY en [Hz])

% TONES
[a1, b1, c1, d1] = FIT_load_test_freq_interruption('5vpp_0.125kHz,14400');
[a2, b2, c2, d2] = FIT_load_test_freq_interruption('5vpp_0.25kHz,14400');
[a3, b3, c3, d3] = FIT_load_test_freq_interruption('5vpp_0.5kHz,14400');
[a4, b4, c4, d4] = FIT_load_test_freq_interruption('5vpp_1.0kHz,14400');
[a5, b5, c5, d5] = FIT_load_test_freq_interruption('5vpp_2.0kHz,14400');
[a6, b6, c6, d6] = FIT_load_test_freq_interruption('5vpp_4.0kHz,14400');
[a7, b7, c7, d7] = FIT_load_test_freq_interruption('5vpp_8.0kHz,14400');
[a8, b8, c8, d8] = FIT_load_test_freq_interruption('5vpp_12.0kHz,14400');
%%
% Frecuencias Fundamentales
%xFF = [c1 c2 c3 c4 c5 c6 c7 c8]'; yFF = [d1 d2 d3 d4 d5 d6 d7 d8]';
FF = [[c1 c2 c3 c4 c5 c6 c7 c8]' [d1 d2 d3 d4 d5 d6 d7 d8]'];
FFMa0 = FF(FF(:,2)>=0,:); %son los pares de puntos mayores a cero
%xFF = FFMa0(:,1)
%yFF = FFMa0(:,2)

clear c1 c2 c3 c4 c5 c6 c7 c8 d1 d2 d3 d4 d5 d6 d7 d8

```

```

%close all
filename = 'el_fiteo.mat'; save(filename);
%% Ajuste de una curva a los maximos de los tonos (FIT)

%cd /Users/matiassandoval/Documents/Universidad-ex_pendrive/Memoria/
% Calibraciones/calibracion_adc_avr32_uc3_frecuencia_interrupcion
%load el_fiteo.mat

figID9 = figure(9);
hold on; grid on; box on;

%smoothingspline or cubicspline
%[curve, goodness, output] = fit(FFMa0(:,1), FFMa0(:,2), 'smoothingspline');
%[curve, P] = csaps(FFMa0(:,1), FFMa0(:,2), 'cubicspline');

%poly3
%[curve, goodness, output] = fit(FFMa0(:,1), FFMa0(:,2), 'poly3');

%gauss2
[curve, goodness, output] = fit(FFMa0(:,1), FFMa0(:,2), 'gauss2');

%poly1
%X = linspace(0, 7250, 80);
%Y = polyval([fit_object.p1 fit_object.p2 fit_object.p3], X);
%h1 = plot(xFF, yFF, 'o--b');
%h2 = plot(X, Y, '-r');

fi = plot(curve,FFMa0(:,1),FFMa0(:,2));

cu1 = plot(a1,b1);
cu2 = plot(a2,b2);
cu3 = plot(a3,b3);
cu4 = plot(a4,b4);
cu5 = plot(a5,b5);
cu6 = plot(a6,b6);
cu7 = plot(a7,b7);
cu8 = plot(a8,b8);

%smoothingspline
%pf = plot(curve,FFMa0(:,1),FFMa0(:,2));

% legend([p8 h2],{'Voltajes leídos en el ADC'; sprintf('Y=(%.4f)X + (%.4f),
% R^2= %.4f',fit_object.p1,fit_object.p2,G.rsquare)},'FontSize',12,
% 'Location','northwest')

title('Funcion de Transferencia y "FIT" de los tonos (Interrupción)');
xlabel('Frequency (Hz)');

```

```

ylabel('Power/Frequency (dB/Hz)');
axis([0,7250,-42,10]); %sine
saveas(figID9,fullfile('graficos_load_test_f','G_freq_int_figura9.png'));

hold off

```

B.5. Code: 'FIT_load_test_freq_interruption.m'

```

%% Extrae y Grafica los datos
% Matias Sandoval y Miguel Martinez.
% 22/06/18
function [a, b, c, d] = FIT_load_test_freq_interruption(filename)
%% Load File Contents
coma1 = strfind(filename,',' );
name = filename(1:coma1-1);
frequency = filename(6:coma1-1);
Fs_adc = str2double(filename(coma1+1:end));

fileID1 = fopen(sprintf('sine_%s.log',name),'r');
fileID2 = fopen(sprintf('tek0000ALL_%s.csv',name),'r');

%for string and floating point number
filecontent1 = textscan(fileID1,'%s', 'Delimiter', '\n');
filecontent2 = textscan(fileID2,'%s', 'Delimiter', '\n');

fclose(fileID1);
fclose(fileID2);

filelines1 = filecontent1{1};
filelines2 = filecontent2{1};
%clear filecontent fileID name2

%% Process File Lines
% Para los archivos 'cutecom' o terminados en '.log'.
inputdata1 = {};
filedata1 = {};
filedata11 = {};
%flag1 = 11; %este es el marcador que indica la fila en la que empiezan los datos
for i = 1:length(filelines1)
%     if strcmp(filelines1{i},...
%     'Micros, HIGH_GAIN, LOW_GAIN, TEMP1, TEMP2, V5, V6 (no se utiliza).')
%         flag1 = i; % lugar en donde esta ésta linea
%     end
%if flag1<= i && i < length(filelines1)

```

```

        %for string and floating point number
    %   filedata1 = textscan(filelines1{i},'%s', 'Delimiter', ' ');
    %   inputdata1{end+1} = filedata1{1}; %filedata{1}(11); esta funcionó
    %end
    if isempty(filelines1{i})==0 %entra si la linea no esta vacia
        %filedata1 = textscan(filelines1{i},'%s', 'Delimiter', ' ');
        filedata1 = textscan(filelines1{i},'%s', 'Delimiter', '['');
        if(strcmp(char(filedata1{1}(1)), 'arreglo_2'))
            filedata11 = textscan(char(filedata1{1}(2)), '%s', 'Delimiter', ' ');
            inputdata1{end+1} = filedata11{1}(2);
        end
    end
end
end

% Para los archivos 'tek0000ALL'.
inputdata2 = {};
filedata2 = {};
flag2 = 22; %este es el marcador que indica cuando empiezan los datos
for i = flag2:length(filelines2)
    %   if strcmp(filelines2{i}, 'Label,,,Label,') || ...
    %       strcmp(filelines2{i}, 'Label,,,Label,,,Label,')
    %       flag2 = i+1; % lugar en donde ésta linea
    %   end
    if flag2 <= i && i < length(filelines2)
        %for string and floating point number
        filedata2 = textscan(filelines2{i}, '%s', 'Delimiter', ',,');
        inputdata2{end+1} = filedata2{1}; %filedata{1}(11); esta funcionó
    end
end

%Lectura de la frecuencia de sampleo del osciloscopio
comas = strfind(filelines2{9}, ',,');
Ts_osc = str2double(filelines2{9}(comas(1)+1:comas(2)-1));
Fs_osc = 1/(Ts_osc);

%clear filedata filelines i
%% Extract Variables
% Para los archivos 'cutecom'.
row_string={};
CH1_string={};
CH2_string={};
CH3_string={};
CH4_string={}; CH4_bin_string={}; CH4_dec_value=[];
CH5_string={}; CH5_bin_string={}; CH5_dec_value=[];
CH6_string={};

for i=1:length(inputdata1)

```

```

values1=inputdata1{i};
%row_string{i}=values1{1}; %ROW
%CH1_string{i}=values1{3}; %HIGH_GAIN
%CH2_string{i}=values1{5}; %LOW_GAIN
%CH3_string{i}=values1{7}; %TEMP1
%CH4_string{i}=values1{9}; %TEMP2
%CH5_string{i}=values1{11}; %V5
%CH6_string{i}=values1{13}; %V6 (no se utiliza)
CH5_bin_string{i}=values1{1}; %ROW

%Load HEX of Channel 4
% [CH4_DEC1, CH4_DEC2] = strtok(CH4_string{i},'||');
% CH4_DEC2 = CH4_DEC2(3:end);
% CH4_BIN1 = dec2bin(str2double(CH4_DEC1),8);
% CH4_BIN2 = dec2bin(str2double(CH4_DEC2),8);
% CH4_bin_string{i} = sprintf('%s%s',CH4_BIN1,CH4_BIN2);
% CH4_dec_value(i) = bin2dec(CH4_bin_string{i});

%Load HEX of Channel 5
% [CH5_DEC1, CH5_DEC2] = strtok(CH5_string{i},'||');
% CH5_DEC2 = CH5_DEC2(3:end);
% CH5_BIN1 = dec2bin(str2double(CH5_DEC1),8);
% CH5_BIN2 = dec2bin(str2double(CH5_DEC2),8);
% CH5_bin_string{i}=sprintf('%s%s',CH5_BIN1,CH5_BIN2);

%Conversiones.
% Counts en complemento 2 (todos positivos) a counts con signo.
% CH5_count = bin2dec(CH5_bin_string{i});
CH5_count = str2double(CH5_bin_string{i});
bits = 16;
if 0 <= CH5_count && CH5_count <= 32767 % 1° recta:
    % valores entre 0 y 2^(bits-1)-1
    CH5_count = CH5_count;
elseif 32768 <= CH5_count && CH5_count <= 65535 %2° recta:
    % valores entre 2^(bits-1) y 2^(bits)-1
    m1 = 1/(1-2^(1-bits)); %pendiente
    CH5_count = m1 * CH5_count + ((1-2^bits) * m1) - 1;
end
% De count con signo a volts.
m2 = 20 / 65536; % 2^16 = 65.536
n2 = 10 - 32767 * m2;
CH5_dec_value(i) = m2 * CH5_count' + n2;
%aquí escojo la columna de datos

end

% Para los archivos 'tek0000ALL'.

```

```

time_string={};
sonda1_string={}; sonda1_dec_value=[];
sonda2_string={};
sonda4_string={}; sonda4_dec_value=[];
sonda5_string={}; sonda5_dec_value=[];
%d_time={};

for i = 1:length(inputdata2)
    values2=inputdata2{i};
    time_string{i}=values2{1}; %TIME
    sonda1_string{i}=values2{2}; %CH1
    sonda2_string{i}=values2{3}; %CH2
    sonda4_string{i}=values2{4}; %FREQUENCY
    sonda5_string{i}=values2{5}; %"MATH<FFT(CH1, HANNING, LOGRMS)>"

    time_dec_value(i) = str2double(time_string{i});
    sonda1_dec_value(i) = str2double(sonda1_string{i});
    sonda4_dec_value(i) = str2double(sonda4_string{i});
    sonda5_dec_value(i) = str2double(sonda5_string{i});
end
%% Create a New Folder
mkdir graficos_load_test_f
%% Grafico de la señal que entrega el osciloscopio
% figID1 = figure(1);
% hold on
% grid on
% box on
% %ax1 = subplot(2,1,1);
% plot(time_dec_value, sonda1_dec_value)
% xlabel('Tiempo [s]')
% ylabel('Voltaje [V]')
% %axis([-0.5,0.5,-1.5,1.5]) %sine
% %axis([-0.5,0.5,-1.5,1.5]) %noise
% title('Señal que entrega el Osciloscopio (Interrupción)')
% saveas(figID1,fullfile('graficos_load_test_f','G_frec_int_figura1.png'))
% hold off
%% Obtencion de la FFT de la señal que entrega el osciloscopio

figID2 = figure(2);
hold on
grid on
box on

L_osc = length(sonda1_dec_value); %largo de los datos
xdft_osc = fft(sonda1_dec_value);
%xFFT_osc = abs(xdft_osc)/N_osc;
xFFT_osc = abs(xdft_osc/L_osc); %lo cambie por un ejemplo en la web

```

```

xdft_osc_mitad = xFFT_osc(1:L_osc/2+1); % obtengo solo la parte positiva
                                     % de la matriz de las frecuencias.
psdx_osc = (xdft_osc_mitad.^2);% * (1/(Fs_osc*N_osc)); % calculo la PSD
psdx_osc(2:end-1) = 2*psdx_osc(2:end-1);

%f = Fs*(0:(L/2))/L;
freq_osc = 0:Fs_osc/L_osc:Fs_osc/2; % la antigua
freq_osc2 = Fs_osc*(0:1/L_osc:1/2);
freq_osc3 = Fs_osc*(0:(L_osc/2))/L_osc;

% voy a acotar los datos hasta 12k[Hz]
cota_12k_osc = find(freq_osc>12450,1);
freq_osc = freq_osc(1:cota_12k_osc);
xdft_osc_mitad = xdft_osc_mitad(1:cota_12k_osc); % estas 4 lineas son para
psdx_osc = psdx_osc(1:cota_12k_osc); % acotar los datos hasta 12.5k[HZ]

p2 = plot(freq_osc,10*log10(xdft_osc_mitad),'-');
xlabel('Frequency (Hz)')
ylabel('Power/Frequency (dB/Hz)')
%legend([p1],{sprintf('%s',name)},'FontSize',12,'Location','northwest')
%axis([0,12500,-100,10]) %sine
title('"Fast Fourier Transform" para la señal del Osciloscopio (Interrupción)')
saveas(figID2,fullfile('graficos_load_test_f','G_freq_int_figura2.png'))
hold off
%% Obtencion de la 'Power Spectral Density Estimates Using FFT' para los
% datos el osciloscopio.

figID3 = figure(3);
hold on
grid on
box on

p3 = plot(freq_osc,10*log10(psdx_osc),'-');
title('"Power Spectral Density" para la señal del Osciloscopio (Interrupción)')
xlabel('Frequency (Hz)')
ylabel('Power/Frequency (dB/Hz)')
%axis([0,12500,-100,10]) %sine
%axis([0,60000000,-140,20]) %noise
saveas(figID3,fullfile('graficos_load_test_f','G_freq_int_figura3.png'))
hold off
%% Grafico de la señal que entrega el ADC

% figID4 = figure(4);
% hold on
% grid on
% box on

```

```

% %ax1 = subplot(2,1,1);
% Ts_adc = 1/Fs_adc;
% tiempo = 0:Ts_adc:(length(CH5_dec_value)-1)*Ts_adc;
% plot(tiempo, CH5_dec_value);%%%%%%%%%%
% xlabel('Tiempo [s]')
% ylabel('Voltaje [V]')
% %axis([0,1200,-1.5,1.5]) %sine
% %axis([0,1200,-1.5,1.5]) %noise
% title('Señal de salida del ADC (Interrupción)')
% saveas(figID4,fullfile('graficos_load_test_f','G_frec_int_figura4.png'))
% hold off
%% Obtencion de la 'Power Spectral Density Estimates Using FFT' para los
% datos del ADC.

figID5 = figure(5);
hold on
grid on
box on

L_adc = length(CH5_dec_value);
xdft_adc = fft(CH5_dec_value);
xFFT_adc = abs(xdft_adc)/L_adc;
xdft_adc_mitad = xFFT_adc(1:L_adc/2+1); % obtengo solo la parte positiva
% de la matriz de las frecuencias.
psdx_adc = xdft_adc_mitad.^2;% * (1/(Fs_adc*N_adc)); % calculo la PSD
psdx_adc(2:end-1) = 2*psdx_adc(2:end-1);

freq_adc = 0:Fs_adc/L_adc:Fs_adc/2;
freq_adc2 = Fs_adc*(0:(L_adc/2))/L_adc;

p5 = plot(freq_adc,10*log10(xdft_adc_mitad));
title('"Fast Fourier Transform" para la señal de salida del ADC (Interrupción)')
xlabel('Frequency (Hz)')
ylabel('Power/Frequency (dB/Hz)')
% axis([0,7250,-95,10]) %sine
% axis([0,42,-20,30]) %noise
saveas(figID5,fullfile('graficos_load_test_f','G_frec_int_figura5.png'));
hold off
%% Obtencion de la 'Power Spectral Density Estimates Using FFT' para los
% datos del ADC.

figID6 = figure(6);
hold on
grid on
box on

p6 = plot(freq_adc,10*log10(psdx_adc));

```

```

title('"Power Spectral Density" para la señal de salida del ADC (Interrupción)')
xlabel('Frequency (Hz)')
ylabel('Power/Frequency (dB/Hz)')
% axis([0,7250,-95,10]) %sine
% axis([0,42,-90,10]) %noise
saveas(figID6,fullfile('graficos_load_test_f','G_frec_int_figura6.png'));
hold off
%% Redimensionar el tamaño del vector mas grande (datos de salida)

% voy a acotar los datos hasta 7.2k[Hz]
cota_7_2k_osc = find(freq_osc>7200,1);
freq_osc = freq_osc(1:cota_7_2k_osc);
xdft_osc_mitad = xdft_osc_mitad(1:cota_7_2k_osc); % estas 4 lineas son para
psdx_osc = psdx_osc(1:cota_7_2k_osc); % acotar los datos hasta 12.5k[HZ]

psdx_adc_redi = [];
m = length(psdx_adc)/length(psdx_osc);
if floor(m)==m % Entra cuando m es entero, o sea son una division exacta.
    % M es el largo del vector de elementos que se tienen que promediar.
    for i=1:length(psdx_osc)
        %for j=1:length(psdx_osc)
            psdx_adc_redi(i) = sum(psdx_adc((i-1)*floor(m):i*floor(m)))/m;
        %end
    end
else %los vectores no son multiplos
    for i=1:length(psdx_osc)
        %for j=1:length(psdx_osc)
            psdx_adc_redi(i) = sum(psdx_adc(((i-1)*floor(m)+1):i*floor(m)))/m;
        %end
    end
end

% antiguas
%ro = sqrt(psdx_adc./psdx_osc_redi);
%transfer_function_1 = (1/2).*(10.*log10(psdx_adc_redi)-10.*log10(psdx_osc));
%transfer_function_2 = -(1/2).*(10.*log10(psdx_adc_redi)-10.*log10(psdx_osc));

% actuales
transfer_function_1 = 10.*log10(sqrt(psdx_adc_redi - psdx_osc));
%transfer_function_2 = -10.*log10(sqrt(psdx_adc_redi - psdx_osc));
%% Calculo de la funcion de transferencia 1

figID7 = figure(7);
hold on
grid on
box on

```

```

p7 = plot(freq_osc,transfer_function_1);

title('Funcion de Transferencia 1(Interrupción)');
xlabel('Frequency (Hz)');
ylabel('Power/Frequency (dB/Hz)');
%axis([0,7250,-42,10]); %sine
saveas(figID7,fullfile('graficos_load_test_f','G_freq_int_figura7.png'));
hold off
%% Calculo de la funcion de transferencia 2

% figID8 = figure(8);
% hold on
% grid on
% box on
% p8 = plot(freq_osc,10.*log10(transfer_function_2));
% title('Funcion de Transferencia 2(Interrupción)');
% xlabel('Frequency (Hz)');
% ylabel('Power/Frequency (dB/Hz)');
% saveas(figID8,fullfile('graficos_load_test_f','G_freq_int_figura8.png'));
% hold off
%% Termino

% Elección de los valores más grandes (frecuencias fundamentales)

% Voy a acotar los datos para encontrar el maximo
y_cota_superior = find(real(transfer_function_1)==max(real(transfer_function_1)),1);
x_cota_superior = freq_osc(y_cota_superior);

% Salidas de la función
a = freq_osc;
b = real(transfer_function_1);
c = x_cota_superior;
d = real(transfer_function_1(y_cota_superior));

% Impresiones
fprintf('Esta lista la medición: ');
fprintf(name);
fprintf('\n');
%close all

```