



Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias Físico Matemáticas

Registro de Eventos de Rayos Cósmicos
Capturados con una Red Pitaya

Tesis

Para obtener el grado de
Licenciado en Física

Presenta:

C. Victor Manuel Acametitla López

Directores de tesis:

Dr. José Rubén Conde Sánchez

Dr. Jorge Cotzomi Paleta

Puebla, Pue.

Julio 2023

Título: Registro de Eventos de Rayos Cósmicos Capturados con una Redpitaya

Estudiante: Victor Manuel Acametitla López

JURADO

Dr. Eduardo Moreno Barbosa
Presidente

Dr. Enrique Varela Carlos
Secretario

Dr. Lucio Fidel Rebolledo Herrera
Vocal

M. en C. Sergio Adán Juárez
Suplente

Dr. José Rubén Conde Sánchez
Director de tesis

Dr. Jorge Cotzomi Paleta
Director de tesis

Agradecimientos

Estoy en deuda con mi familia por tan grande apoyo y paciencia que me han ofrecido a lo largo de estos años. En especial a mis padres, Marcos Magdaleno Acametitla Osorio y Margarita López Rojas, por darme la oportunidad de seguir con mis estudios de licenciatura. A mis hermanos Marcos, Perla y Magaly por ofrecerme ese gran apoyo emocional.

A mis asesores el Dr. José Ruben Conde Sanchez y el Dr. Jorge Cotzomi Paleta por brindarme su apoyo y paciencia durante estos últimos años.

A los sinodales encargados de la revisión de este trabajo, Dr. Eduardo Moreno Barbosa, Dr. Lucio Rebolledo, M.C. Sergio Adán Juárez y Dr. Enrique Varela Carlos por su gran contribución como jurado en mi tesis. Su experiencia, conocimiento y orientación han sido fundamentales en el desarrollo de este trabajo.

A mis preciados amigos Miguel, Julio, Luis, Ariadna, Anita, Diana y Rocío por estar presentes a lo largo de mi estadia en la facultad.

Índice general

Resumen	1
1. Introducción	3
1.1. Planteamiento del problema	4
1.2. Justificación	6
1.3. Objetivo general	6
1.4. Objetivos específicos	6
2. Antecedentes	9
2.1. Rayos cósmicos	9
2.2. Espectro de energía	9
2.3. Origen de rayos cósmicos primarios	11
2.4. Composición química de los rayos cósmicos	12
2.5. Cascadas atmosféricas de partículas	13
2.5.1. Fenomenología básica	13
2.6. Atmósfera terrestre	17
2.6.1. Profundidad atmosférica	18
3. Instrumentación	21
3.0.1. Detectores de centelleo	21
3.0.2. SiPM micro FJ-60035 SensL	27
3.0.3. Fotomultiplicadores de silicio	27
3.0.4. Sistemas previos de adquisición de datos	33
3.1. Redpitaya	34
3.1.1. Tarjeta STEMLab 125 – 14	35
3.1.2. Características	35

3.1.3.	Resolución del convertidor analógico digital	36
3.1.4.	<i>Firmware</i>	37
3.2.	Vivado	41
3.2.1.	Esquema de los módulos	42
3.2.2.	Módulo del sistema de procesamiento	42
3.2.3.	Módulo de la adquisición de datos	43
3.2.4.	Módulos de la memoria FIFO	44
3.2.5.	Módulos AXI	44
3.2.6.	Modos para la Adquisición de datos	46
3.2.7.	Arreglo experimental	47
4.	Resultados	51
4.0.1.	Pruebas a la instrumentación	51
4.0.2.	Adquisición de datos con Generador de funciones tektronix AFG310 y Generador de Pulsos Aligent 81101A	52
4.0.3.	Ruido generado por STEMLab 125 – 14	57
4.0.4.	Perfiles de trazas a 16 muestras de un SiPM	59
4.0.5.	Perfiles de trazas a 32 muestras de un SiPM	60
4.0.6.	Perfiles de trazas a 64 muestras de un SiPM	61
5.	Conclusiones y trabajo a futuro	67
5.0.1.	Perspectivas futuras:	67
	Apéndice	69
5.1.	Enlace de conexión entre una tarjeta Redpitaya y una PC	69
5.1.1.	Scripts en python	75
6.	*Apendice 3	79
6.0.1.	Principales <i>scripts</i> en VHDL	79

Índice de figuras

2.2.1.Espectro de energía de RC	10
2.4.1.Abundancia de elementos en rayos cósmicos y el sistema solar	13
2.5.1.Esquema de la estructura general de una EAS generada por un RC primario	15
2.6.1.Esquema del perfil de temperaturas de la atmósfera como función de la altitud	18
3.0.1.Estructura interna del SiPM y símbolo de los componentes. El SiPM consiste en un conjunto de microcélulas conectadas en paralelo. Cada microcélula incluye un APD, una resistencia y un condensador rápido CF tomado de [13].	28
3.0.2.Diagrama del arreglo electrónico conformado por SiPM's, amplificador y Red Pitaya.	29
3.0.3.Esquema del funcionamiento del amplificador, tomado de [10]	29
3.0.4.Lectura del pulso de salida estándar mostrando el tiempo de recuperación.	30
3.1.1.Redpitaya	34
3.1.2.Esquema de Tarjeta Redpitaya modelo STEMLAB 125-24	35
3.1.3.Diagrama de flujo de la implementación de memoria FIFO	39
3.2.1.Esquema memoria FIFO	41

3.2.2. Esquema completo de la configuración de módulo en vivado.	42
3.2.3. Módulo <i>processing system</i> en vivado.	43
3.2.4. Módulo adquisición de datos en vivado.	44
3.2.5. Módulo control de la memoria FIFO en vivado.	44
3.2.6. Arreglo formado por el generador de pulsos Agilent 81101A, la tarjeta Redpitaya STEMLab 125-14 y una laptop.	48
3.2.7. Pulso emitido por el generador donde podemos observar que la tarjeta puede almacenar y visualizar la misma señal.	48
3.2.8. Captura de los perfiles de un Pulso emitido por el generador donde podemos observar que la tarjeta puede almacenar y visualizar la misma señal.	49
3.2.9. Arreglo experimental donde están conectados el SiPM, fuente de voltaje que lo alimenta, STEMLab 125-14 y una computadora.	49
3.2.10. Pulso emitido por un SiPM visto desde un osciloscopio tektronik 3020.	50
3.2.11. SiPM sin la cubierta de color negro, al lado de STEMLab 125-14.	50
4.0.1. Comprobación del Modo FreeCounter, como resultado del conteo de muestras por STEMLab 125-14.	51
4.0.2. Comprobación del Modo de trazas del contador de 16 muestras por STEMLab 125-14.	52
4.0.3. Comprobación del Modo de trazas del contador de 32 muestras por STEMLab 125-14.	52
4.0.4. Visualización a diferentes señales de prueba.	53
4.0.5. Visualización a diferentes tamaños de muestras de un Pulso a 100kHz.	54
4.0.6. Visualización a diferentes tamaños de muestras de un Pulso a 5 MHz.	55

4.0.7. Visualización de diferentes pulsos muestreados con el modo FreeCascada.	56
4.0.8. Visualización de diferentes perfiles de pulsos muestreados con el modo TrazaCascada.	57
4.0.9. Niveles del ruido tanto en unidades adimensionales así como en voltaje.	58
4.0.10. Histogramas de las distribuciones del ruido donde STEM-lab 125-14 está con y sin instrumentos extras visualizado en voltaje (mV).	59
4.0.11. Perfiles de pulsos detectados por un SiPM así como la visualización de las 16 muestras de cada uno en unidades adimensionales.	59
4.0.12. Visualización sobre algunos de los perfiles de pulsos detectados por un SiPM en el rango de 640 a 800 dentro del almacenamiento de 1024 muestras.	60
4.0.13. Perfiles de pulsos detectados por un SiPM así como la visualización de las 32 muestras de cada perfil en unidades adimensionales.	61
4.0.14. Visualización sobre algunos de los perfiles de pulsos detectados por un SiPM en el rango de 300 a 500 dentro del almacenamiento de 512 muestras.	61
4.0.15. Perfiles de pulsos detectados del SiPM a 64 muestras.	62
4.0.16. Visualización sobre algunos de los perfiles de pulsos detectados por un SiPM en el rango de 190 a 255 dentro del almacenamiento de 512 muestras.	62
4.0.17. Diferentes conjuntos de perfiles de pulsos adquiridos por la Redpitaya.	63
4.0.18. Histograma de amplitud de los perfiles almacenados.	64
4.0.19. Histograma de carga.	65
5.1.1. Vista de la conexión local.	70
5.1.2. Estructura de Redpitaya	71

5.1.3. Interfaz en <i>powershell</i> para acceder al entorno de la tarjeta.	71
5.1.4. Interfaz Web de Redpitaya.	72
5.1.5. Menú de <i>Development</i>	73
5.1.6. Conexión establecida con la tarjeta.	73
5.1.7. Selección de la opción <i>Python programming</i> para manipular el control del <i>firmware</i> de la adquisición de datos.	74

Resumen

Los rayos cósmicos son partículas de $\sim 10^{20}$ eV de energía que viajan a velocidades cercanas a la luz y que en su viaje, al interactuar con la atmósfera terrestre, provocan una lluvia de partículas secundarias que pueden ser captadas en Detectores de superficie. Para estudiar los rayos cósmicos se requiere de instrumentos que digitalizan la señal y que operen a la velocidad del tiempo de vida de los eventos del orden de $\sim 10^{-9}$ s. La presente tesis contribuye al estudio de los rayos cósmicos por medio de: (1) Desarrollar programas para digitalizar y almacenar datos empleando una Redpitaya que opera a 125 MSPS¹ (8ns) y resolución en amplitud de 14 bits. (2) Adquisición y análisis de señales de rayos cósmicos con Redpitaya registrados en un SiPM, ubicado a una altura de 2160 m.s.n.m (3) Resultados y conclusiones de las señales adquiridas.

¹Del inglés Mega Sample Per Second.

Capítulo 1

Introducción

En el año de 1912, Victor Hess descubrió una radiación que tenía naturaleza extraterrestre, otorgándole el Premio Nobel de Física en 1936 al medir el incremento de radiación ionizante en función de la altura mientras se realizaba un ascenso en globo. Dicha radiación fue definida como rayos cósmicos primarios los cuales consisten en partículas cargadas que llegan a la tierra de forma interestelar con altos niveles de energía mayores a 10^9 eV y que pueden alcanzar niveles de $\sim 10^{20}$ eV.

En el vasto universo que nos rodea, existen fenómenos cósmicos de gran envergadura y misterio, capaces de desencadenar eventos de una magnitud inigualable. Entre estos fenómenos, los rayos cósmicos se presentan como una fuente de energía y partículas altamente energéticas provenientes del espacio exterior. Su origen y naturaleza continúan siendo objeto de estudio e investigación en la comunidad científica.

Los rayos cósmicos, compuestos principalmente por partículas cargadas, como protones y núcleos atómicos, viajan a velocidades cercanas a la de la luz y pueden alcanzar energías extremadamente elevadas. Estos eventos son generados por diversos procesos astrofísicos, como explosiones de supernovas, agujeros negros supermasivos o incluso eventos aún desconocidos. A medida que estos rayos ingresan a la atmósfera terrestre, interactúan con los átomos y moléculas, produciendo una cascada de partículas secundarias que atraviesan el

aire y la materia circundante.

La detección y el registro preciso de los eventos de rayos cósmicos son fundamentales para comprender la naturaleza y el origen de estas partículas de alta energía. Además, estos datos proporcionan información valiosa sobre la física de partículas, la estructura del universo y la radiación ionizante presente en nuestro entorno.

En este contexto, el presente trabajo de tesis se centra en el registro de eventos de rayos cósmicos capturados por detectores especialmente diseñados. El objetivo principal es desarrollar un sistema de adquisición y análisis de datos que permita detectar y registrar con precisión la llegada de los rayos cósmicos y analizar sus características energéticas y temporales. Esto implicará la utilización de instrumentación moderna y tecnología avanzada, así como la implementación de algoritmos y técnicas de procesamiento de datos.

La importancia de este estudio radica en el avance del conocimiento científico en el campo de la astrofísica y la física de partículas. Además, los resultados obtenidos podrían tener aplicaciones prácticas en áreas como la monitorización de radiación espacial, la protección de equipos electrónicos sensibles y la evaluación de riesgos en entornos donde la exposición a radiación ionizante es relevante.

A lo largo de este trabajo de tesis, se explorarán diferentes aspectos relacionados con la detección y el registro de eventos de rayos cósmicos. Se describirán los fundamentos teóricos y conceptuales que respaldan este campo de investigación, así como las metodologías experimentales y técnicas de procesamiento de datos utilizadas. Además, se presentarán los resultados obtenidos a través de experimentos y simulaciones, y se discutirán sus implicaciones y posibles extensiones.

1.1. Planteamiento del problema

Frecuentemente, los rayos cósmicos primarios colisionan a lo largo de sus trayectorias con partículas de la atmósfera terrestre, lo que da

lugar a la producción de partículas secundarias como muones positivos, muones negativos, piones, entre otros. Este fenómeno es conocido como cascadas aéreas extensas o cascadas atmosféricas de partículas.

Cuando el rayo cósmico primario posee suficiente energía al llegar a la atmósfera terrestre, las partículas secundarias generadas en la cascada atmosférica alcanzan un nivel de energía detectable. Estas partículas interactúan con ciertos detectores, por ejemplo, los WCD¹, donde la respuesta es una señal eléctrica y esta es procesada por un convertidor analógico-digital (ADC).

Estas cascadas de partículas tienen tiempos de vida cortos, en el orden de 10^{-9} segundos, lo que requiere que la instrumentación utilizada para detectar estos eventos sea rápida y precisa. **Sin embargo, dado que el sistema actual lleva implementado más de una década, ha comenzado a presentar algunos problemas. El principal inconveniente se encuentra en el ADC, el cual debe ser producido y montado exclusivamente para la adquisición de datos de un WCD.** A medida que pasa el tiempo, esto implica que se requiere más tiempo para producir y probar estos componentes, así como realizar las reparaciones necesarias cuando sea necesario. Además, el costo de mantenimiento del sistema de adquisición de datos se vuelve cada vez más elevado, ya que los componentes de fabricación han sido descontinuados y complica encontrar repuestos.

Ante las oportunidades identificadas, es necesario emplear instrumentación moderna que facilite las actualizaciones en el funcionamiento en hardware a través de la modificación de su software y que sea adaptable a experimentos de mayor complejidad. Dado que la detección de rayos cósmicos ocurre en la superficie terrestre, los detectores deben enfrentar las condiciones climáticas externas. Por

¹Water Cherenkov Detector

consiguiente, la electrónica utilizada debe contar con características que le permitan resistir entornos adversos. En este sentido, la incorporación de dispositivos programables versátiles se vuelve crucial, ya que son capaces de detectar estos eventos fenomenológicos dentro de los tiempos establecidos, ofreciendo una mayor resistencia para operar durante períodos prolongados.

1.2. Justificación

Para abordar este problema, en este trabajo de tesis se propone la utilización de la tarjeta Redpitaya, basada en un SoC, modelo STEMLab 125 – 14 como sistema de adquisición de datos. Una de las principales ventajas de esta tarjeta es la presencia de una entrada rápida de doble ADC y una salida rápida de doble DAC. Con una frecuencia de muestreo de 125 MSPS, la configuración estándar de la tarjeta permite almacenar diferentes ventanas temporales. Esto significa que Redpitaya puede digitalizar datos cada 8 ns, lo cual la sitúa en el rango adecuado para detectar pulsos con una duración del orden de 10^{-9} segundos. Además, actualmente hay una escasez de instrumentación comercial de bajo costo para la adquisición de señales de fenómenos físicos ultrarrápidos, lo que hace que la implementación de Redpitaya sea oportuna en este sentido.

1.3. Objetivo general

Desarrollar e Implementar estructuras que contribuyan al registro de eventos de Rayos Cósmicos captados con una Redpitaya.

1.4. Objetivos específicos

- Desarrollar programas para digitalizar y almacenar datos empleando una Redpitaya que opera a 125 MSPS y resolución de 14 bits.
- Evaluar la funcionalidad y el desempeño del sistema de adquisi-

ción mediante la realización de pruebas utilizando instrumentos como un generador de pulsos, un generador de funciones y un detector.

- Verificar la capacidad del sistema para adquirir, procesar y registrar de manera precisa y confiable las señales provenientes de diferentes fuentes como lo son un generador de pulsos y un fotodetector.

La presente tesis esta constituida por los siguientes capítulos:

El primer capítulo (Capítulo 1) consiste en una breve introducción donde se aborda el planteamiento del problema, la justificación para el uso de la tarjeta programable, así como los objetivos planteados. En el segundo capítulo (Capítulo 2), se ofrece una introducción a la física de la radiación cósmica y se explora el origen de los rayos cósmicos. Además, se proporciona una descripción general de la fenomenología de las lluvias de partículas producidas por la interacción de los rayos cósmicos primarios con la atmósfera terrestre, junto con detalles generales de las capas que la componen.

En el tercer capítulo (Capítulo 3), se aborda el tema de la instrumentación involucrada, incluyendo las características generales del SiPM en conjunto con el plástico centellador, así como el funcionamiento del detector. También se profundiza en las características de la Redpitaya STEMLab 125-14, la implementación de la memoria FIFO y los códigos desarrollados tanto para la implementación en hardware con VHDL como en hardware con Python.

En el cuarto capítulo (Capítulo 4) se presentan los resultados obtenidos en la adquisición de datos de un generador de pulsos y del fotodetector, así como los histogramas obtenidos a partir de estos datos.

Capítulo 2

Antecedentes

2.1. Rayos cósmicos

Los rayos cósmicos que golpean la atmósfera de la Tierra son núcleos ionizados, que aproximadamente un 90 % son protones, un 9 % de partículas alfa y el resto núcleos más pesados, electrones y neutrones. La mayoría de los rayos cósmicos son relativistas, con altas energías, donde una pequeña fracción de ellos que llegan hasta los 10^{20} eV de energía (alrededor de 20 J) [1].

Históricamente, el origen extraterrestre de los rayos cósmicos fue descubierto por Victor Hess en 1912, cuando midió el aumento de la radiación ionizante con la altura durante los ascensos en globo [5].

2.2. Espectro de energía

El flujo de rayos cósmicos que llegan a la tierra se determina aproximadamente por una ley de potencias de la forma

$$\left(\frac{dN}{dE}\right) \approx E^{-\gamma}. \quad (2.1)$$

Donde N representa el número de partículas, E la energía de las partículas y γ el índice espectral. Así mismo, el valor de γ en el espectro de energía representa dos cambios importantes en la ley de potencia, identificados como la rodilla y tobillo, y cuyos valores están

asociados al origen de los rayos cósmicos, galácticos y extragalácticos. La figura 2.2.1 ofrece una visión general del espectro completo de los rayos cósmicos, donde se observa un rango suficientemente amplio de energías. y se resaltan características importantes en el espectro, siendo la rodilla(*knee*) a 10^{15} eV y el tobillo(*ankle*) a 10^{18} eV [3].

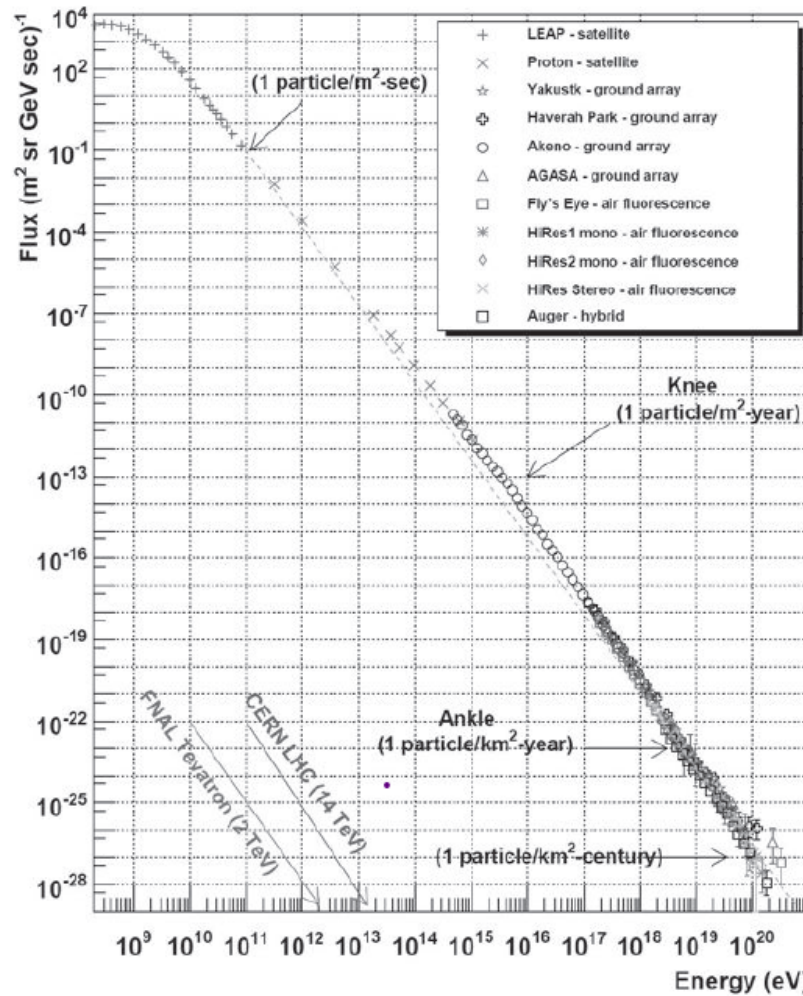


Figura 2.2.1: Flujo de rayos cósmicos en función del espectro energético, donde se indican las características más destacadas, la rodilla a 10^{15} eV y tobillo a 10^{18} eV. Además, se indica la frecuencia de llegada de algunas partículas a diferentes energías [3].

Además, el espectro global de energía se puede dividir en cuatro regiones de 10^9 eV a 10^{15} eV el índice espectral diferencial es $\gamma \approx$

$-2,7$. a partir de 10^{15} a 10^{18} eV el valor es de $-3,1$. Para niveles de energía arriba de los 10^{18} eV el espectro se vuelve a aplanar hacia $\gamma \approx -2,6$. y luego aparentemente sufre un corte cerca de los 10^{20} eV [1].

Es conveniente considerar primero los rayos cósmicos con energías en el rango de 10^9 a 10^{15} eV y posteriormente los de energías superiores. Esta división corresponde a las diferentes técnicas e instrumentos que se utilizan para la detección los rayos cósmicos, los detectores de partículas de los observatorios espaciales abarcan el rango de energía más bajo y las técnicas de chubascos de aire de rayos cósmicos en las energías más altas [3].

2.3. Origen de rayos cósmicos primarios

Las emisiones de rayos cósmicos pueden ser producidas por varios procesos físicos, y esto se deduce de acuerdo con sus energías detectadas. Las partículas con baja energía, por debajo de los 100 MeV (10^8 eV), en su mayor parte son repelidas en el espacio exterior siendo esta la razón de su poca presencia a bajos niveles de energía dentro del espectro de energía. A su vez, los rayos cósmicos con energía de 100 MeV (10^8 eV) se genera en el interior del sistema solar, en su mayor parte irradiada por el Sol; el resto de la radiación, por arriba de 1 GeV (10^9 eV) aproximadamente, procede del exterior del sistema solar y se propagan sin perturbaciones apreciables [4].

De manera local la variación periódica más importante se da por el ciclo solar cada 11 años y la intensidad de la radiación cambia con las manchas solares. Sin embargo, la porción del espectro que abarca hasta los 10^{15} eV, lo que corresponde a la “rodilla” se debe a partículas de origen galáctico, Como posibles causas de aceleración se les da a eventos masivos como supernovas; son los eventos de mayor energía que se han observado en el universo, además cuando ocurren

estas explosiones, se producen la mayoría de los elementos pesados encontrados en el universo y dispersándolos en su entorno [4].

Los rayos cósmicos a energías superiores a el “tobillo” que se producen entre 10^{18} y 10^{19} eV son de origen extra galáctico siendo las posibles causantes las galaxias con núcleo activo (AGN’S) [4]. Estas galaxias son los objetos más luminosos y violentos en el universo, esto debido a que puede se asume que en su centro poseen un agujero negro supermasivo que eyecta cantidades colosales de gas y polvo.

2.4. Composición química de los rayos cósmicos

La composición y abundancia química de los rayos cósmicos proporciona datos importantes sobre su origen y sobre los procesos de propagación desde las fuentes que los generan hasta la superficie terrestre y estas pueden ser comparadas entre la abundancia de elementos que se producen en el sistema solar con la abundancia de elementos interestelares [3].

La composición química de la radiación cósmica es relativamente constante en el rango de energía desde unos 10^6 eV hasta los 10^9 eV y la mayoría de los elementos encontrados aquí se observó que no hay diferencias considerables entre la abundancia hallada en los rayos cósmicos galácticos y los producidos dentro del sistema solar [4].

La Figura 2.4.1 muestra las similitudes y diferencias generales entre las abundancias de los rayos cósmicos, así como abundancia típica de elementos del sistema solar e interestelar. Así, podemos apreciar que los elementos ligeros como el litio (Li), el berilio (Be) y el boro (B) son sobre abundantes en los rayos cósmicos en relación con su abundancia en el sistema solar [3].

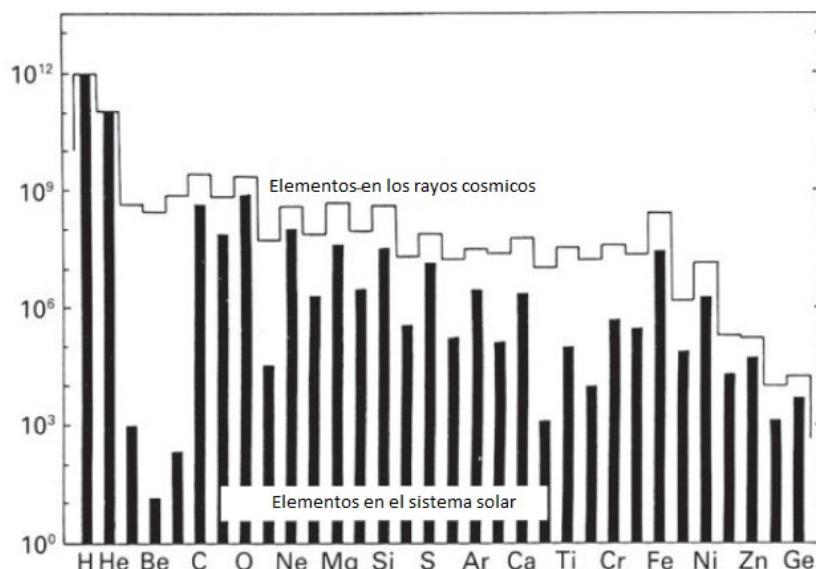


Figura 2.4.1: Abundancia de elementos en los rayos cósmicos (línea continua) comparada con la abundancia en el sistema solar (histograma sólido). Los datos fueron normalizados a una abundancia relativa de hidrogeno a 10^{12} (Lund, 1984) [3].

2.5. Cascadas atmosféricas de partículas

2.5.1. Fenomenología básica

Cuando los rayos cósmicos primarios que entran de manera isotrópica al planeta, interactúan con las partículas que componen la atmósfera y se producen grandes flujos de partículas secundarias, las cuales se dan por una gran cantidad de colisiones sucesivas con los elementos que componen a la atmósfera, a su vez, se debe considerar que la composición química de la atmósfera en mayor medida se trata de 78 % nitrógeno (N_2), 21 % oxígeno (O_2) y el resto por practicidad, se toma en cuenta al argón (Ar) como el 1 % restante. A este fenómeno se le da el nombre de Cascadas Extensas de Partículas (denominadas del inglés como EAS).

De los procesos de interacción donde se producen las partículas secundarias. Los describimos brevemente:

Ionización y excitación

Las partículas cargadas que viajan a través de la materia pierden energía al excitarse e ionizar los átomos a medida que colisionan con ellos. Esto ocurre cuando la colisión con un electrón de un átomo es lo suficientemente energética como para liberar un electrón de su estado actual, lo que resulta en la formación de un electrón libre y la creación de un ion cargado positivamente.

Radiación de Bremsstrahlung

La palabra "Bremsstrahlung", de origen alemán, significa radiación de frenado. Este tipo de radiación se produce cuando una partícula cargada con masa interactúa con el campo eléctrico de un núcleo atómico, lo que resulta en una desaceleración de la partícula y la consiguiente pérdida de energía. Esta energía perdida se emite como radiación electromagnética en forma de fotones de bremsstrahlung.

Creación de pares

La creación de pares electrón-positrón es un proceso en el cual las partículas de alta energía interactúan con la materia y producen simultáneamente un electrón y un positrón. Esta interacción ocurre cuando se transfiere la energía suficiente para materializarse en forma de un par de partículas con cargas opuestas. Este fenómeno se observa cuando un fotón de alta energía interacciona con un campo electromagnético y se convierte en un par de partículas con igual masa. Este proceso se rige por el principio de conservación de energía, y la energía necesaria para que ocurra es $E = 1,022 \text{ MeV}$.

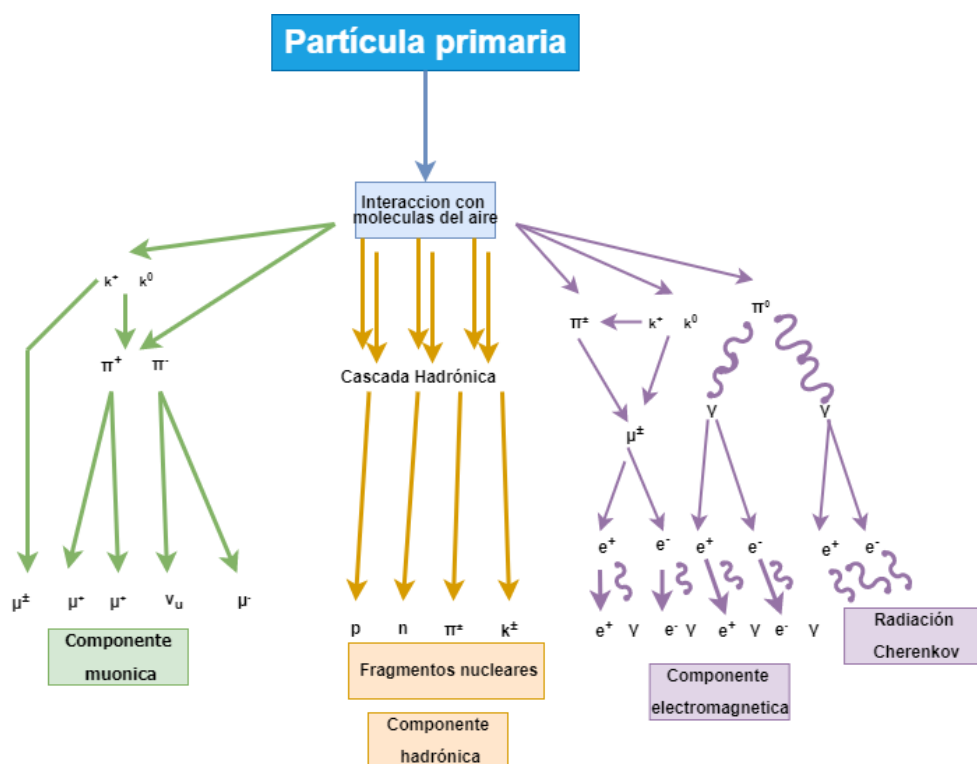


Figura 2.5.1: Esquema de la estructura general de una EAS generada por un RC primario, produciendo tres componentes: Hadrónica, muónica y electromagnética.

Las mediciones que se obtienen de las cascadas se pueden desarrollar en términos de la energía y la composición química del rayo cósmico primario [8]. Además, se clasifican en tres componentes: hadrónica, muónica y electromagnética. En la figura 2.5.1 podemos observar un esquema de las componentes que se generan por la interacción de un rayo cósmico primario con algún núcleo que conforma a la atmósfera.

Cascada Hadrónica

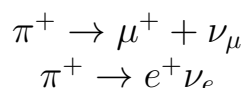
La mayoría de las cascadas de esta componente son iniciadas por hadrones extremadamente energéticos con energías superiores a 10^{13} eV [6], y se fragmenta en la primera interacción que se produce a mayor altura que aquellas producidas por protones, creados en su gran mayoría durante las primeras interacciones hadrónicas de los

rayos cósmicos en la atmósfera. Cuanto más energético es la partícula incidente, los fragmentos resultantes de la espalación continúan propagándose y creando una gran cantidad de interacciones sucesivas, los secundarios más abundantes que se producen en las primeras interacciones son: piones π^\pm , kaones k^\pm y partículas cargadas a lo largo de su trayectoria por la atmósfera.

También, estos últimos se encuentran en el núcleo de la cascada, cercano al eje de la lluvia y en consecuencia poseen un nivel de dispersión lateral pequeño no mayor a algunos cientos de metros.

Cascada Muónica

Los muones y neutrinos se crean principalmente durante el decaimiento de piones cargados y kaones de las primeras interacciones en la cascada hadrónica principal, este proceso de decaimiento de piones cargados se produce a través de la siguiente reacción.



Cascada Electromagnética

Los piones neutros (π^0) y en menor cantidad los muones (μ^\pm) abren canales electromagnéticos en el momento que decaen y desvían una parte significativa de la energía de la cascada principal hadrónica hacia rayos gamma (γ) y electrones de alta energía (e^+, e^-), estos a su vez, en un proceso iterativo, generan un mayor número de cascadas electromagnéticas. Por esta razón, el comportamiento de las EAS queda completamente determinado por la componente electromagnética [6].

En general se puede decir que las cascadas de partículas a nivel del mar, aproximadamente el 90 % son electrones y positrones de energías relativamente bajas, cerca del 10 % son muones y menos del 1 % se trata de hadrones [6].

2.6. Atmósfera terrestre

La atmósfera terrestre consiste en un gran volumen de gas con una densidad de aproximadamente 10^9 partículas por cm^3 a nivel del mar. Con el aumento de la altitud, la densidad del aire disminuye y con ella también el número de moléculas y núcleos, como mencionamos en apartados anteriores, la atmósfera está compuesta en su mayoría por nitrógeno y oxígeno. Con frecuencia se utiliza una representación aproximada denominada *atmósfera exponencial isotérmica estándar*.

En la *homosfera* es la composición de tres regiones donde la composición atmosférica permanece generalmente constante. Las cuales son: *Troposfera*, *estratosfera* y *mesosfera*. Estas regiones específicas se definen en función de sus variaciones de temperatura.

La *troposfera* es la capa donde tienen lugar los procesos resultantes de los fenómenos meteorológicos y donde se concentra hasta el 80 % de la masa total.

La *estratosfera*, que generalmente carece de nubes y en la que se concentra el ozono, así en esta capa que va de los 16 a 50 km de altitud, el calentamiento debido a la absorción de la radiación ultravioleta por las moléculas de ozono es responsable del máximo de temperatura aproximadamente a los 50 km.

La *mesosfera*, situada entre 50 y 80 km, en la que la temperatura disminuye al aumentar la altitud.

La *termosfera*, en la que la temperatura aumenta con la altitud hasta unos 130 km de altitud. Debido a la absorción de la radiación solar asociado a la separación de molecular diatómicas de O_2 y N_2 .

El perfil de temperatura de la atmósfera en función de la altitud

se muestra en la Fig 2.6.1. Las capas entre las diferentes regiones se denominan pausas, es decir, tropopausa, estratopausa, mesopausa y termopausa.

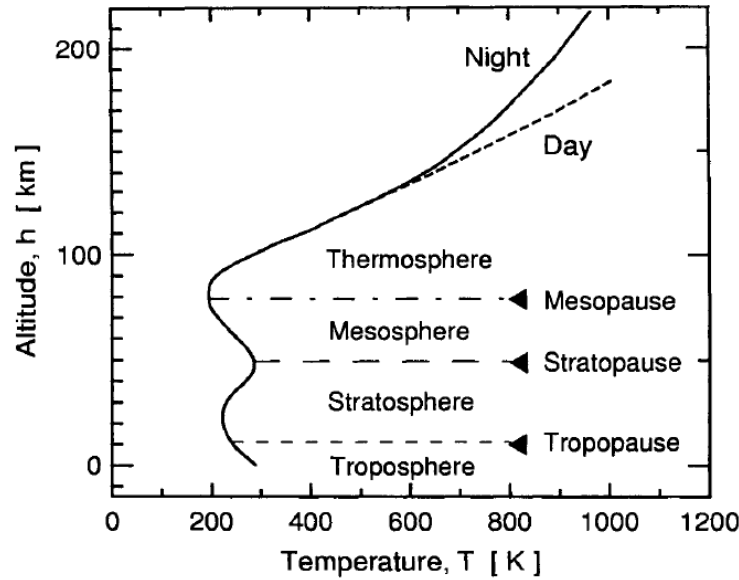


Figura 2.6.1: Esquema del perfil de temperaturas de la atmósfera como función de la altitud [6].

2.6.1. Profundidad atmosférica

El parámetro principal para describir las interacciones producidas en la atmósfera se trata de la cantidad de materia con la que la partícula primaria puede interactuar, producir partículas secundarias e iniciar cascadas de partículas. Dicha cantidad es la denominada profundidad atmosférica X , con unidad de medida en g/cm^2 . La profundidad atmosférica es definida como la integral de la densidad atmosférica en función de la altitud (h), expresada como:

$$X = \int_0^{h_1} \rho(h) \cdot dh. \tag{2.2}$$

En donde, $\rho(h_1)$ representa la densidad de la atmósfera a una altura h_1 [8], siendo ρ una cantidad importante que determina la relación entre las interacciones y el decaimiento de partículas.

Como la relación entre la presión atmosférica (siendo equivalente a X) y la densidad de la atmósfera es proporcional a la temperatura. Y, si la temperatura fue constante la relación entre la altitud y la profundidad es de la forma

$$X = X_0 \exp -h/h_0. \quad (2.3)$$

Donde X_0 es la profundidad atmosférica al nivel del mar, esto es, 1030 g/cm^3 y h_0 es la escala de altura de la atmósfera. Siendo verdadero para un gas perfecto con una composición constante y en equilibrio hidrostático.

Capítulo 3

Instrumentación

La electrónica requerida en la instrumentación consiste de lo siguiente:

- Plástico Centellador.
- SiPM (Fotomultiplicador de Silicio).
- Tarjeta Redpitaya STEMLab 125-14.
- Desarrollo de programas tanto en lenguaje hardware (FPGA) como en software (Framework de python), lo que constituye el *firmware*.

3.0.1. Detectores de centelleo

El uso de materiales capaces de producir luz centelleante es una de las técnicas inicialmente empleadas para la detección de radiación ionizante. En particular, los plásticos centelleadores se fabrican disolviendo un centelleador orgánico en un solvente y luego polimerizándolo. Esto da lugar a la formación de un equivalente sólido de la solución.

Los materiales adecuados para ese objetivo deben tener las siguientes propiedades:

- Deben convertir la energía cinética de partículas cargadas en luz detectable con alta eficiencia de centelleo.

- Este proceso de conversión deberá ser lineal, de esta forma se tiene que el rendimiento de luz deba ser proporcional a la energía depositada en el mayor rango posible.
- El material debe ser transparente para las longitudes de onda emitidas por el mismo y se obtenga una aceptable adquisición de luz.
- El tiempo de decaimiento de la luz inducida debe ser corto para generar pulsos rápidos.
- Contar con un índice de refracción cercano al cristal que es aproximadamente 1,5 para hacer posible un buen funcionamiento y la luz centellante sea captada por cualquier modelo de tubo fotomultiplicador.

Centelladores orgánicos

Mecanismos

El proceso de fluorescencia consiste en la emisión instantánea de radiación visible de una sustancia después de producirse una excitación por algún medio. Por otro lado, la fosforescencia corresponde a la emisión de luz de longitud de onda más larga que el primer proceso, además tiene un tiempo característico que es más lento.

La fluorescencia en materiales orgánicos se produce a través de transiciones en la estructura de niveles de energía de una sola molécula, lo que permite su observación en un tipo específico de molécula, independientemente de su estado físico [7].

Muchos centelladores se basan en moléculas orgánicas con propiedades de simetría que generan estructuras de π -electrones. Estas moléculas pueden absorber energía y excitarse en varios estados. En el caso de los centelladores orgánicos, la diferencia energética entre los estados s_0 y s_1 es de alrededor de 3 ó 4 eV. Además, la eficiencia de centelleo se define como: la fracción de la energía total de la

partícula incidente que se convierte en luz visible. Sin embargo, existen procesos de desexcitación en los que no se emite luz y la energía excitada se disipa en forma de calor, a estos procesos se les denomina *quenching*. En la mayoría de los materiales orgánicos, se produce una transferencia significativa de energía entre moléculas antes de la pérdida de excitación [7].

Tipos de centelladores orgánicos

Dentro de esta categoría se cuentan los cristales orgánicos puros, soluciones orgánicas sólidas y los plásticos centelladores.

Cristales orgánicos puros

El antraceno es un material ampliamente utilizado con fines de centelleo debido a su alta eficiencia de centelleo, lo que significa que produce una mayor salida de luz por unidad de energía. Por otro lado, el estilbeno se utiliza para la discriminación de pulsos entre centelleos inducidos por partículas cargadas y electrones, aunque su eficiencia es menor en comparación con el antraceno. Ambos materiales son frágiles y difíciles de obtener en grandes cantidades. Es importante destacar que la eficiencia de centelleo también depende de la orientación de la partícula ionizante con respecto al eje del cristal. Esto puede resultar en una resolución energética insuficiente, con una reducción de hasta el 30% en estos cristales. En cuanto a las dimensiones, los cristales utilizados tienen un diámetro y grosor máximo de menos de 10 cm [7].

Soluciones líquidas

Estos centelladores se producen disolviendo un centellador orgánico en un disolvente, y a veces se agrega un tercer compuesto para ajustar su respuesta espectral a los tubos fotomultiplicadores (PMT) comunes. A diferencia de los cristales y plásticos, los centelladores líquidos no tienen una estructura sólida que pueda dañarse por exposición intensa. Por lo tanto, se espera que sean más resistentes a

los efectos de la radiación [7].

Plásticos centelladores

Estos centelladores se producen al disolver un centellador orgánico en un disolvente que puede polimerizarse posteriormente, lo que resulta en una solución sólida. Un disolvente común utilizado es el monómero de estireno, en el cual se disuelve el centellador orgánico. Luego, el estireno se polimeriza para formar un plástico sólido. También se puede utilizar poliviniltolueno o polimetilmetacrilato como disolvente [7].

La ventaja de utilizar plásticos como centelladores es la facilidad de fabricación, permite producir una amplia variedad de tamaños. Esto permite adaptarlos a diversas aplicaciones y requerimientos específicos.

Respuesta de los centelladores orgánicos

La luz de salida

Una porción de la energía cinética perdida por una partícula cargada en un centellador se transforma en energía fluorescente, lo que se conoce como *eficiencia de centelleo*, mientras que el resto de la energía se disipa en forma de vibraciones o calor.

La eficiencia de centelleo depende tanto del tipo de partícula como de su energía. En algunos casos, esta eficiencia puede ser independiente de la energía, lo que resulta en una dependencia lineal del rendimiento lumínico en relación con la energía inicial [7].

En centelladores líquidos y plásticos, la respuesta a electrones es lineal para energías superiores a 125 keV. Sin embargo, la respuesta a partículas cargadas más pesadas, como protones o partículas alfa, es siempre menor en comparación para energías equivalentes, y su comportamiento no es lineal para energías mucho mayores [7].

Para describir el rendimiento lumínico absoluto de estos centella-

dores, se utiliza la nomenclatura MeV (equivalente de electrones). Esto significa que se compara la energía necesaria para generar 1 MeV de luz. Por definición, se requiere 1 MeV de energía para generar 1 MeV de luz en el caso de electrones rápidos. Sin embargo, para partículas cargadas más pesadas, se necesitan varios MeV de energía para generar 1 MeV de luz, debido a su menor rendimiento lumínico por unidad de energía [7] .

Tiempo de respuesta

Dado que los estados luminiscentes en una molécula orgánica se forman de manera instantánea y solo se observa fluorescencia inmediata, el perfil temporal del pulso de luz emitido debe tener un flanco de subida muy rápido seguido de un decaimiento exponencial simple.

Sin embargo, un análisis más detallado de la dependencia temporal del rendimiento de centelleo debe tener en cuenta otros dos efectos. En primer lugar, se necesita un tiempo finito para poblar los estados luminiscentes, lo cual puede requerir tiempos cercanos a medio nanosegundo para que se produzca la fluorescencia inmediata. En segundo lugar, existen componentes más lentos del centelleo que corresponden a la fluorescencia y fosforescencia retardadas [7].

En los centelladores de gran tamaño, la fluctuación en el tiempo de tránsito de la luz debido a múltiples reflexiones en las superficies del centellador puede generar diferencias significativas en los tiempos de llegada de la luz a los sensores de los detectores. Esto implica que la fluorescencia juega un papel importante en el aparente empeoramiento de la resolución temporal a medida que aumentan las dimensiones de un centellador [7].

Centelladores inorgánicos

Los centelladores inorgánicos suelen estar compuestos por un sólo cristal de sal alcalina, como el NaI. Al ser un monocristal, tienen la

Detección de radiación	BC-404	BC-408
Alphas, Betas	x	x
Partículas cargadas, rayos cósmicos, muones, etc.	x	x
Propiedades de centelleo		
Salida de luz % antraceno	68	64
Tiempo de subida, ns	0.7	0.9
Tiempo de caída (ns)	1.8	2.1
Ancho del pulso, ns	2.2	-2.5
Longitud de onda máximo, emisión, nm	408	425
Longitud de atenuación de la luz	160	380

Tabla 3.0.1: Especificaciones de los centelladores BC-404, BC-408

capacidad de ser transparentes, lo que evita la reflexión y absorción en las caras del cristal. La presencia de reflexiones y absorciones reduciría la eficacia del centellador.

Las interacciones entre el cristal y los átomos provocan que los niveles de energía aumenten, lo que da lugar a la formación de una serie de bandas de energía. Esta serie de bandas incluye una banda de valencia y una banda de conducción, que son las dos bandas superiores en el espectro energético [7].

Características del plástico centellador implementado

Las dimensiones físicas del plástico centellador utilizado son de 30 mm de grosor con un diámetro de 18 cm. Actualmente, se dispone de poca información sobre él, ya que no ha sido caracterizado debido a que fue fabricado en el laboratorio de materiales de la Benemérita Universidad Autónoma de Puebla. No obstante, podemos compararlo con la respuesta general de otro plástico centellador del mercado, en concreto el modelo BC404 y BC 404 [11], ya que muestra una similitud en las sus características principales de respuesta. Ver Tabla 3.0.1.

Esto con el propósito de que tanto la respuesta del plástico centellador, como el SIPM coinciden en operar en el mismo rango de longitud de onda permitiendo una la mayor eficiencia de detección posible.

3.0.2. SiPM micro FJ-60035 SensL

La fotoabsorción en el silicio ocurre cuando un fotón atraviesa el material y es absorbido, transfiriendo energía a un electrón y creando un par electrón-hueco. Tomando en cuenta que la profundidad de absorción de los fotones en el silicio depende de su energía o longitud de onda, lo que hace que el silicio sea un material adecuado para la detección de luz a diferentes longitudes de onda [13].

Los fotodiodos de silicio que forman parte del SiPM, consisten en una unión p-n que crea una región de agotamiento libre de portadores de carga móviles, lo que significa que no hay electrones ni huecos libres presentes en esta región. Los iones de dopaje positivo en la zona n y los iones de dopaje negativos en la zona p crean un campo eléctrico que empuja a los portadores de carga hacia sus respectivas regiones, creando así una barrera para el flujo de corriente [13].

Una vez que se genera un campo eléctrico lo suficientemente alto en la región de agotamiento, los portadores de carga se aceleran y adquieren la energía cinética necesaria para crear pares de carga secundarios mediante la ionización por impacto. Durante este proceso, las partículas cargadas interactúan entre sí y transfieren su energía, lo que resulta en la formación de iones. Por lo tanto, cuando un fotón es absorbido, puede desencadenar una cascada de ionización en todo el volumen del silicio sometido a un campo eléctrico. Cuando el silicio se rompe y se vuelve conductor, el primer par electrón-hueco se amplifica y genera un flujo de corriente medible [13].

3.0.3. Fotomultiplicadores de silicio

Los SiPM o fotomultiplicadores de silicio son detectores de radiación con alta sensibilidad, eficiencia y baja fluctuación de tiempo. Están basados en diodos p y n polarizados que detectan la luz en el rango del ultravioleta cercano al infrarrojo cercano. Estos detec-

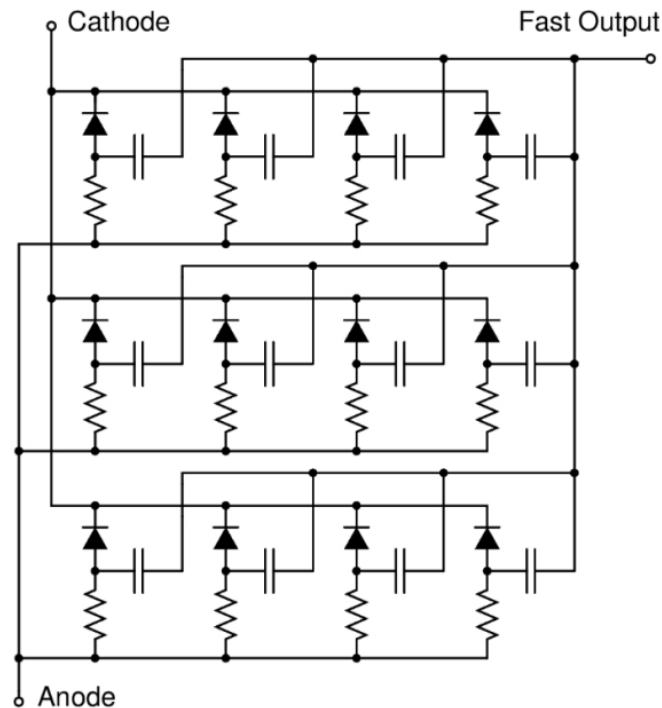


Figura 3.0.1: Estructura interna del SiPM y símbolo de los componentes. El SiPM consiste en un conjunto de microcélulas conectadas en paralelo. Cada microcélula incluye un APD, una resistencia y un condensador rápido CF tomado de [13].

tores están compuestos por una matriz de microceldas sensibles a la luz, que son fotodiodos de avalancha (APD's). En la figura 3.0.1 podemos observar la estructura interna de un SiPM que consta de 12 APD's. Durante este trabajo, en la Figura. 3.0.2 se puede observar el esquema electrónico enfocado en la parte del fotodetector que fue usado, compuesto por cuatro SiPM's conectados en paralelo, y a su vez la corriente emitida pasa por un amplificador LMH 5401 que proporciona una ganancia de 7.

Cuando una microcelda del SiPM se activa en respuesta a la absorción de un fotón, se produce una avalancha de Geiger que genera una corriente fotoeléctrica a través de la microcelda. Esta corriente resulta en una disminución del voltaje a través de la resistencia de *quenching* reduciendo la polarización por debajo del nivel de la

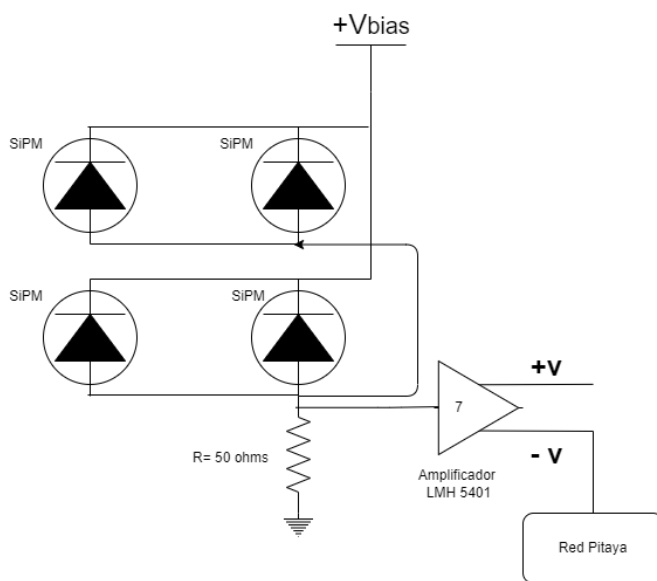


Figura 3.0.2: Diagrama del arreglo electrónico conformado por SiPM's, amplificador y Red Pitaya.

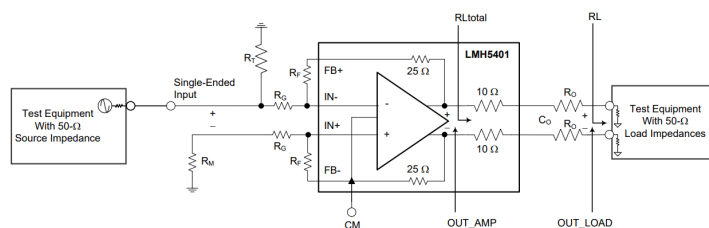


Figura 3.0.3: Esquema del funcionamiento del amplificador, tomado de [10]

resistencia de enfriamiento, apagando así la corriente fotoeléctrica y evitando la generación de nuevas avalanchas en modo Geiger. Una vez que la corriente fotoeléctrica se apaga, el voltaje a través del diodo se recarga hasta alcanzar el valor de polarización nominal.

El diagrama 3.0.3 corresponde al amplificador que es usado junto con el arreglo del detector para aumentar la señal producida por la combinación de cuatro SiPM's. El tiempo que tarda la microcelda en recargarse completamente hasta el voltaje de funcionamiento se denomina tiempo de recuperación. Es importante tener en cuenta que la avalancha de Geiger se limitará a la microcelda en la que se inició. Durante el proceso de avalancha, las restantes microceldas

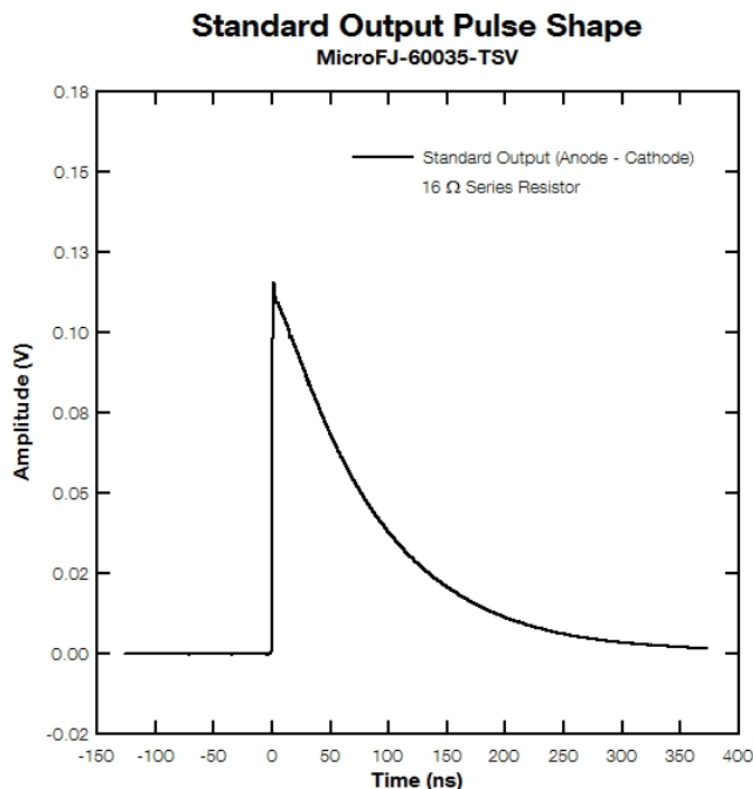


Figura 3.0.4: Lectura del pulso de salida estándar mostrando el tiempo de recuperación.

permanecerán completamente cargadas y listas para detectar fotones.

Cada microcelda detecta fotones de manera idéntica e independiente, y la suma de las corrientes fotoeléctricas de estas microceldas individuales se combina para formar una salida cuasianáloga. Por lo que, el SiPM es capaz de proporcionar información sobre la magnitud de un flujo instantáneo de fotones [13].

El SiPM tiene dos salidas: una salida ultra rápida con un ancho de pulso de 6 ns y una salida estándar con un ancho de pulso cercano a los 400 ns. En la Figura 3.0.4 podemos observar el pulso de la salida estándar.

La salida del sensor es una corriente fotoeléctrica, y la carga total

Q generada por un evento se calcula mediante la ecuación:

$$Q = N_{fired}Gq \quad (3.1)$$

Donde N_{fired} es el número de microceldas disparadas, G es la ganancia y q es la carga del electrón. La carga total también se puede obtener mediante la integral del pulso de corriente fotoeléctrica.

En cuanto a los parámetros de rendimiento del SiPM tenemos los siguientes:

Voltaje de ruptura

El voltaje de ruptura (V_{br}) es el punto de polarización en el que el campo eléctrico en la región de agotamiento es lo suficientemente intenso como para generar una descarga de Geiger. Los SiPM funcionan en un punto de polarización que suele estar entre un 10 % y un 25 % por encima de V_{br} . La diferencia entre V_{br} y el punto de polarización se conoce como sobrevoltaje (ΔV).

Ganancia

La ganancia en un SiPM se define como la cantidad de carga creada por cada fotón detectado y depende del sobrevoltaje y del tamaño de la microcelda. Cada microcelda genera una cantidad uniforme y cuantizada de carga cada vez que se produce una avalancha debido a la absorción de un fotón en el volumen activo de la celda. El funcionamiento único del SiPM permite medir con precisión la ganancia. Cada fotón detectado por el sensor genera un pulso de salida.

Eficiencia de detección de fotones (PDE)

La PDE (*Photon Detection Efficiency*) es la probabilidad estadística de que un fotón incidente interactúe con una microcelda para producir una avalancha. Se define como:

$$PDE(\lambda, V) = \eta(\lambda) * \epsilon(V) * F. \quad (3.2)$$

Parametro	sobrevoltaje (+6V)
PDE	50 %
Tasa de conteo oscuro	150 kHz/mm ²
Ganancia	6.3x10 ⁶
Corriente oscura	7.5 μ A
Constante de tiempo de recarga de microcelda	50 ns
Dimensiones	6.13x6.13 mm ²
Número de microceldas	22292

Tabla 3.0.2: Datasheet MicroFJ-60035.

Donde $\eta(\lambda)$ es la eficiencia cuántica del silicio, que mide la probabilidad de que un fotón incidente genere un par electrón-hueco en el volumen sensible del sensor. $\epsilon(V)$ es la probabilidad de iniciar una avalancha y F es el factor de llenado del dispositivo, que representa la relación entre la superficie activa y la inactiva.

Tasa de conteo oscuro

La principal fuente de ruido en un SiPM es la tasa de conteo oscuro (DCR, Dark Count Rate), que se debe a los electrones térmicos generados en el volumen activo. Cada conteo oscuro es el resultado de un electrón generado térmicamente que inicia una avalancha en la región de campo alto. Estas señales resultantes de la ruptura de la microcelda son consideradas como ruido en la medición. Algunas de estas características podemos verlas en la tabla 3.0.2. Las características generales del modelo en encuentran en [12].

Funcionamiento del detector

Dado el plástico centellador, el cual es un tipo de material que tiene la propiedad de emitir luz cuando es irradiado por partículas en trabajo conjunto con un SiPM, la detección se realiza de la siguiente manera:

- La radiación incide en el plástico centellador y deposita energía en él. Este absorbe dicha energía y la convierte en luz visible mediante el proceso de centelleo.

- Los fotones de luz generados durante el centelleo son detectados por el SiPM. Al recibir los fotones de luz, genera una corriente eléctrica proporcional a la intensidad de la luz detectada.
- La corriente eléctrica generada por el SiPM se amplifica por la avalancha de Geiger y se procesa electrónicamente para su posterior análisis y registro.
- La combinación de un plástico centellador con un SiPM permite detectar y cuantificar la radiación incidente. El SiPM, al ser altamente sensible y tener una respuesta rápida, es capaz de detectar incluso bajos niveles de luz emitida por el plástico centellador. Esto hace posible la detección de radiación con alta sensibilidad y resolución.

3.0.4. Sistemas previos de adquisición de datos

La digitalización de fenómenos físicos ha sido una tarea compleja en los últimos años. En los primeros enfoques, se utilizaban dispositivos modulares que empleaban sistemas de comunicación para transferir datos, como el módulo v1751. Este módulo, desarrollado por la empresa CAEN, es una unidad VME (Versa Module Eurocard) que cuenta con 8 canales de conversión analógico-digital (ADC) de 10 bits y una velocidad de muestreo de 1 MSample/s. Cada canal posee un búfer de memoria con un tamaño que varía entre 1.885 y 14 MS/Channel.

La necesidad de actualizar el sistema de adquisición de datos implementado en los Detectores de Agua Cherenkov (WCD, por sus siglas en inglés) se debía principalmente a mejorar la capacidad de registrar pulsos rápidos (del orden de nanosegundos) detectados por los fotomultiplicadores. El sistema existente contaba con una arquitectura estándar en su electrónica.

El sistema anterior consistía en los siguientes elementos:

- Una tarjeta FPGA Spartan 3E, modelo $XC3S500E - FG230$ de Xilinx. Esta tarjeta utilizaba módulos programados en VHDL (lenguaje de descripción de hardware para circuitos integrados de muy alta velocidad) y se encargaba del proceso de conteo de los pulsos. Por esta razón, se la denominaba tarjeta madre. También contaba con una interfaz en serie para la comunicación con la computadora y la transmisión de datos.
- Una tarjeta digitalizadora, también conocida como tarjeta hija, diseñada a medida. Esta tarjeta contaba con dos convertidores analógico-digital AD9216 de 10 bits y 40 MHz, así como cuatro amplificadores operacionales dedicados a la adquisición de datos.
- Una computadora con sistema operativo Linux conectada mediante un puerto serie.

3.1. Redpitaya

Es una plataforma de hardware de código abierto que se compone principalmente de un *system on chip* combina características de una computadora, un osciloscopio y un generador de señales, todo en un dispositivo de tamaño reducido. Tiene diversas aplicaciones, como adquisición de datos, procesamiento de señales, control y automatización, entre otras.

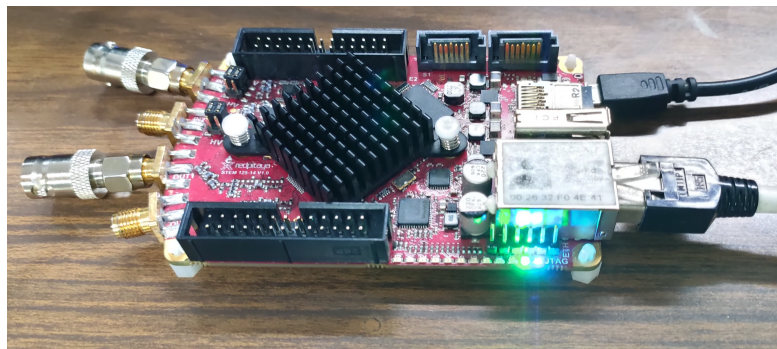


Figura 3.1.1: Tarjeta STEMLAB 125-24.

3.1.1. Tarjeta STEMLab 125 – 14

La tarjeta utilizada para la instrumentación es la correspondiente al modelo STEMLab 125-14, tiene características particulares mostradas en la figura 3.1.2.

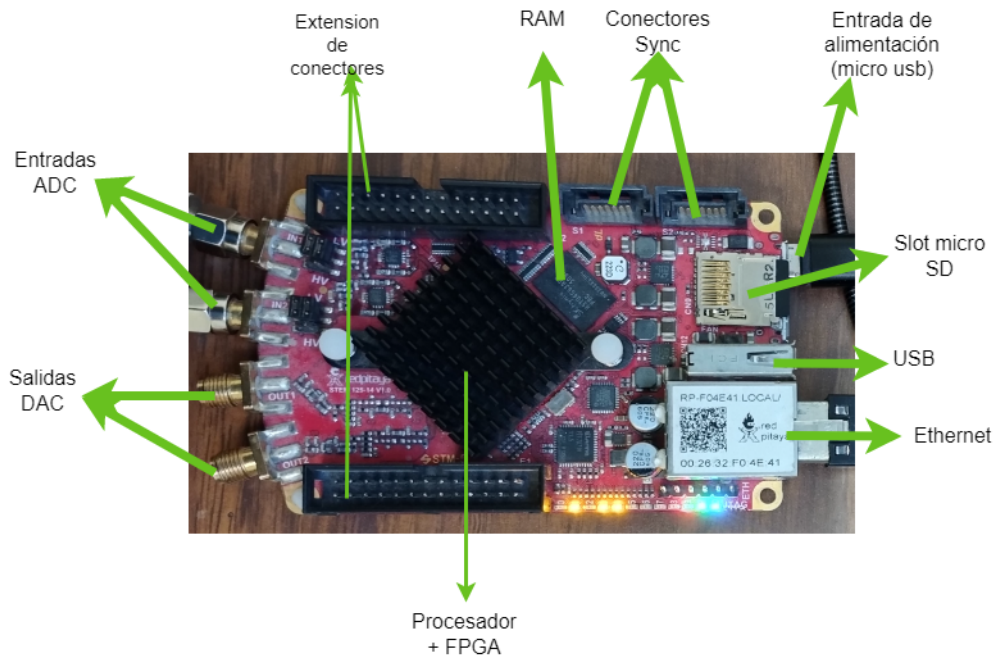


Figura 3.1.2: Esquema de tarjeta Redpitaya modelo STEMLAB 125-24.

3.1.2. Características

La placa de desarrollo usada como núcleo central de la instrumentación tiene las siguientes características globales:

- Procesador de arquitectura ARM modelo Cortex–A9 de doble núcleo de alta velocidad a 1 GHz.
- Almacenamiento interno de datos con una capacidad a partir de 4 GB a una máxima 32 GB por medio de un slot microSD y 512 MB de memoria RAM.

Entradas analógicas	Descripción
Canales de conversión ADC	2
Resolución del ADC	14 bits
Resolución temporal	8 ns
Frecuencia de muestreo	125 MSPS
Conectores	SMA
Voltaje de entrada máxima absoluta	30 V
Protección contra sobre carga	Si
Rango de voltaje de entrada LV	± 1 v
Rango de voltaje de entrada HV	± 20 v
Ancho de banda	50 Mhz

Tabla 3.1.1: Especificaciones de STEMlab 125-14.

- Osciloscopio de alta definición que permite visualización y análisis en tiempo real de señales analógicas y digitales.
- Generador de señales que crea pulsos de alta precisión e interfaz de usuario gráfica intuitiva y fácil acceso.
- Puerto USB.
- Voltaje de alimentación de 5V a 2 Amp máximo con entrada tipo microUSB.
- Puerto Ethernet con conector RJ45.

Para propósito de la tesis se hizo uso sólo de un canal ADC para la entrada analógica, esto permitió la digitalización de los pulsos. La interfaz analógica cuenta con dos canales analógicas de entrada de alta velocidad y cuentan con las siguientes especificaciones, ver tabla 3.1.1.

3.1.3. Resolución del convertidor analógico digital

En general, la resolución de un ADC se representa en número de bits, y establece el número de divisiones que puede tener un rango de entrada analógica. Dado que STEMlab 125-14 tiene una resolución de 14 bits y el rango de voltaje que va de $-1V$ a $1V$, y se expresa en número de bits, entonces la resolución de operación viene dado por

$$\frac{1}{2^n} = \frac{1}{2^{14}} = \frac{1}{16384} = 0,00006103. \quad (3.3)$$

Dado que el rango de entrada es de $2\text{ V} = 2000\text{ mV}$.

$$\frac{2000\text{mV}}{2^{14}} = 0,122070\text{mV}. \quad (3.4)$$

Por tanto, podemos medir variaciones de 0.122070 mV provenientes de la señal de entrada.

3.1.4. *Firmware*

El *firmware* es un tipo de software específico diseñado para ser instalado en microcontroladores, chips o tarjetas electrónicas, que en este caso se instala en el *Sistem on Chip* de la Redpitaya. A diferencia del software de aplicación, que se ejecuta en un sistema operativo, el *firmware* se encuentra integrado en el hardware y tiene acceso directo a sus componentes y funciones.

El *firmware* propuesto tiene la función de controlar y gestionar el funcionamiento interno de la Redpitaya. Proporciona las instrucciones y módulos necesarios para controlar los distintos componentes y periféricos del dispositivo, como el procesador, la memoria, los puertos de entrada/salida, entre otros. También puede incluir configuraciones, protocolos de comunicación y comandos de inicio.

La ventaja de desarrollar el *firmware* es que puede ser actualizado o modificado a medida que se realizan mejoras o correcciones. Esto se logra generando una nueva versión del *firmware*, compilándola y cargándola en la Redpitaya. El módulo clave en el *firmware* desarrollado es la sincronización de la lectura/escritura de la memoria FIFO¹, dada la naturaleza de esta clase de memoria resulta adecuada para almacenar datos a la velocidad de 125 MSPS (8ns).

Memoria FIFO

La interfaz de la memoria FIFO funciona como un registro de desplazamiento de datos, ejecutándose cada vez que se existe un evento

¹First Input-First Output

en reloj, produce un corrimiento de los registros almacenados en la memoria. Estos valores ocupan un espacio físico en el hardware de almacenamiento y, a medida que ocurren las transiciones, se genera un flujo de datos a medida que avanza el reloj. Esto significa que el primer dato que se escribe en la memoria FIFO será el primero en leerse. Esto implica que hay un retraso entre la entrada y la salida de los datos.

Cuando se activa el proceso de escritura y se toman datos a la velocidad de 125 MSPS, se utiliza la señal del clock de valor 1, denominada clock fast. Por otro lado, cuando se activa el proceso de lectura a la velocidad del equipo de cómputo, se utiliza la señal del clock de valor 0, denominada clock low. Es importante destacar que la velocidad de lectura siempre será mucho mayor que la velocidad de escritura. Esta característica se puede observar en la Figura 3.2.1.

En la Figura 3.1.3 se muestra el diagrama de flujo que representa el proceso lógico de la memoria FIFO. Este proceso dinámico de cambio entre la escritura y lectura de datos es posible gracias a las señalizaciones de control presentes en la FIFO.

Para escribir los datos en la memoria, se utiliza la variable 'write_enable' (*wr_en*). Este proceso de escritura se lleva a cabo mientras $wr_en = 1$, es decir, hasta que se llene todo el espacio disponible en la memoria. Una vez que la memoria está llena, se activa una señal de bandera llamada *full*.

A partir de este punto, Redpitaya cambia de instrucción y comienza el proceso de lectura. Se utiliza la variable 'read_enable' (*rd_en*) para este proceso. En este punto, wr_en toma el valor de 0 y rd_en se asigna el valor de 1. Con este cambio de asignación, se deshabilita la escritura y se inicia la lectura de todos los registros almacenados en la memoria.

Una vez que se completa la lectura de todos los registros, se activa otra bandera llamada *empty*. Esta bandera indica que todos los registros han sido leídos. En este punto, los datos pueden ser guardados para su posterior análisis y luego se reinicia la interfaz de la memoria FIFO para repetir este ciclo de instrucciones de manera cíclica.

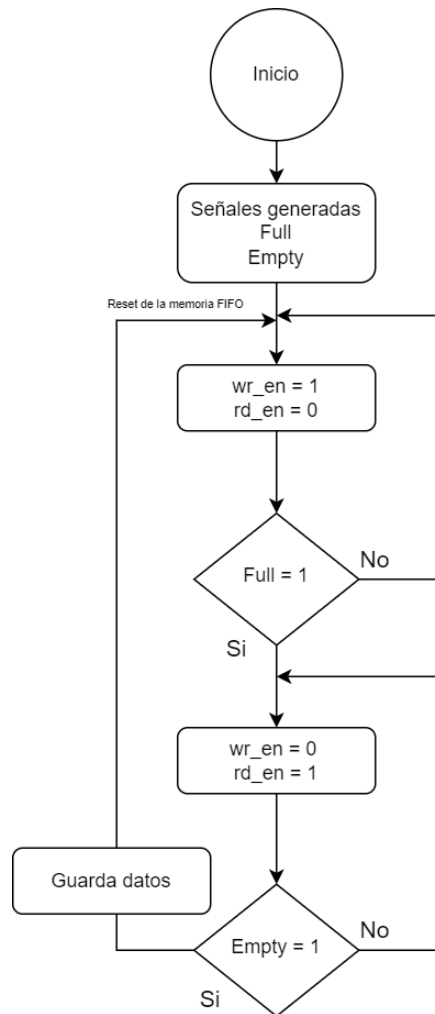


Figura 3.1.3: Diagrama de flujo de la implementación de memoria FIFO.

El segmento de código en lenguaje VHDL que se encarga de la ejecución de la memoria FIFO se trata del código que compone al módulo

de 'CONTROL' el cual lo vemos descrito en el script 6.1:

Cuando la memoria FIFO emite la señalización de la bandera *empty*= 1, la lectura se realiza a la velocidad del framework de Python. El código encargado de esta tarea se muestra a continuación en el *script* 5.1. Su función principal es configurar el umbral de disparo, el cual se ajusta manualmente en función de las características particulares de los pulsos a estudiar. El segundo segmento 5.2 se encarga de extraer los datos de la memoria FIFO.

3.2. Vivado

El esquema mostrado en la Figura 3.2.1 representa el sistema completo de funcionamiento de la interfaz de la memoria FIFO, incluyendo el multiplexor encargado de seleccionar la velocidad del reloj según el modo de operación: lectura o escritura en la memoria FIFO. Este esquema fue desarrollado utilizando la plataforma de desarrollo Vivado de Xilinx para programar la FPGA. Vivado es un entorno de diseño de sistemas integrados (DSI) que ofrece varias ventajas, como la compatibilidad con el lenguaje VHDL y herramientas de diseño, simulación y verificación, síntesis de hardware, implementación y programación de dispositivos. Es especialmente adecuado para la programación de dispositivos como Redpitaya, que contiene un procesador Zynq.

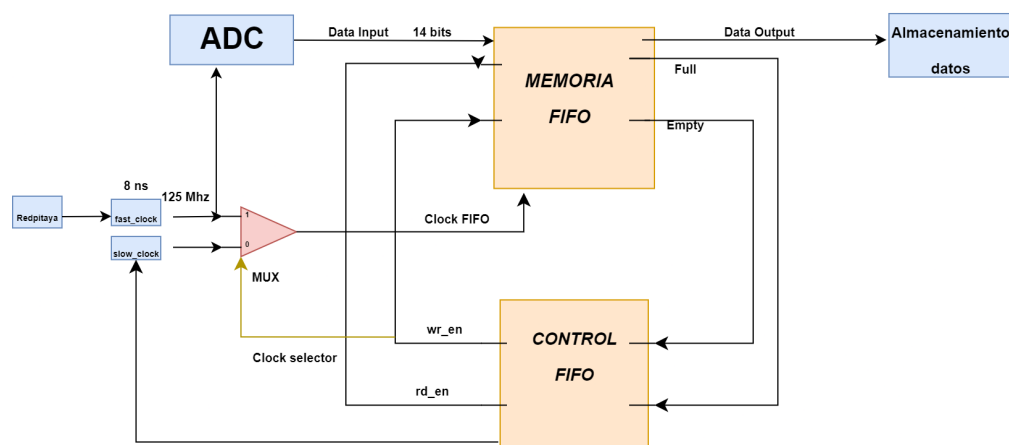


Figura 3.2.1: Esquema completo del funcionamiento de la memoria FIFO.

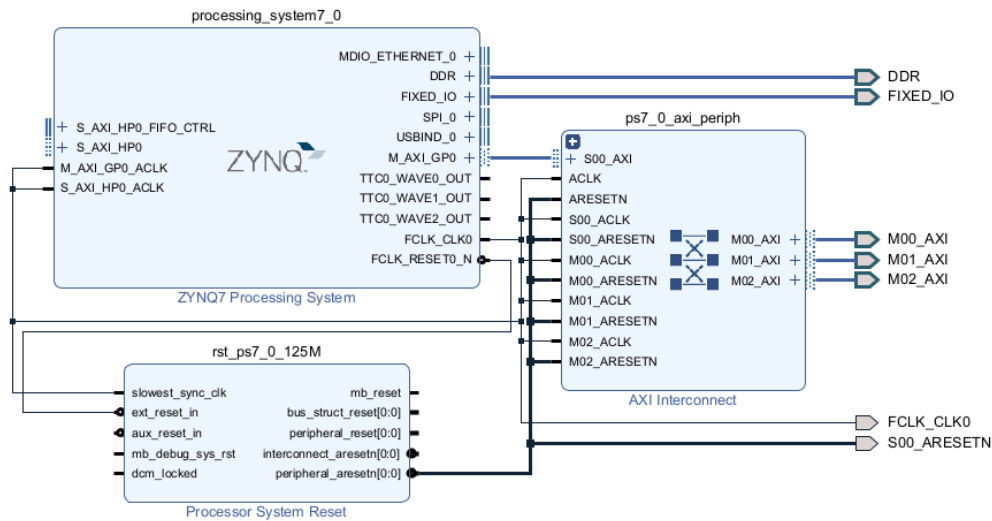


Figura 3.2.3: Módulo *processing system* en vivo.

3.2.3. Módulo de la adquisición de datos

Los módulos involucrados en la adquisición de datos, especialmente los convertidores analógico a digitales, son componentes diseñados para capturar y procesar las señales de entrada en un sistema digital cuyos datos provienen de alguna fuente externa, como sensores o detectores.

El primer módulo en este conjunto, nombrado como `axis_red_pitaya_adc`, cuenta con dos características particulares. Primero convierte el reloj externo de 50 MHz a un reloj interno de 125 MHz usado en la entrada `adc_clk_a` del puerto externo a la lógica programable interna a el reloj `adc_clk`. Como segundo punto, lee los datos del ADC del canal de entrada y los pone a disposición por medio de la interfaz AXI `M_AXIS`. El segundo módulo es `signal_split_0`, transforma la interfaz de salida ADC `M_AXIS` en interfaces de salida `M_AXIS`.

Base Address	AXI	Size
0x4120_0000	AXI_control	64k
0x4121_0000	AXI_umbral	64 k
0x4200_0000	AXI_gencontrol	4 k

Tabla 3.2.1: Direcciones físicas de los módulos AXIS.

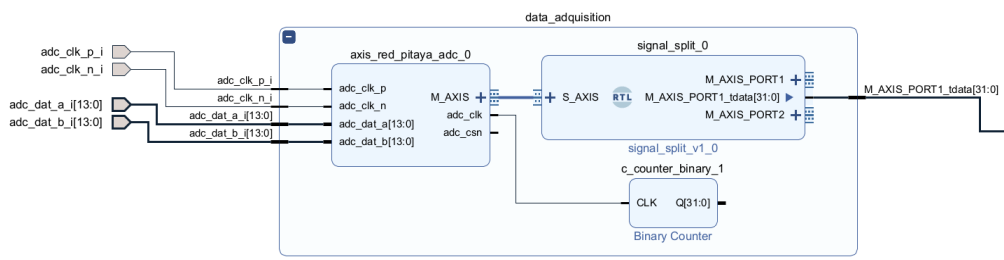


Figura 3.2.4: Módulo adquisición de datos en vivo.

3.2.4. Módulos de la memoria FIFO

Dentro de los módulos importantes en el control de la memoria FIFO tenemos los módulo AXIS

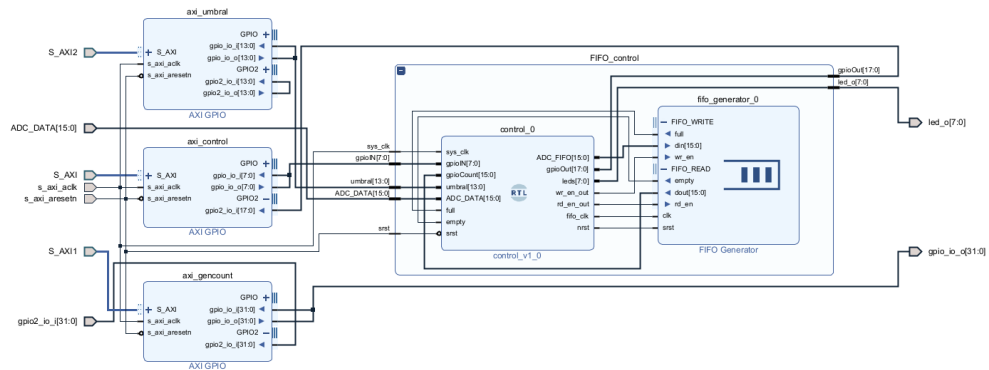


Figura 3.2.5: Módulo control de la memoria FIFO en vivo.

3.2.5. Módulos AXI

Dentro del entorno de desarrollo, es posible agregar elementos denominados AXIS (*Advanced eXtensible Interface Specification*), por sus siglas en inglés, estos módulos encargados de facilitar la comunicación entre diferentes componentes y periféricos en sistemas de

hardware y software. Estos módulos permiten la transferencia de datos de manera serializada a través de canales dedicados llamados *streams*, los cuales suelen ser unidireccionales que son periféricos conectados al procesador. Se generan diferentes tipos de AXIs, como el AXIs de control, encargado de la manipulación de la memoria FIFO, el AXIs de control_umbral, que se encarga del nivel de disparo para capturar trazas, y el AXIs_gentcount, que es un generador de cuentas. La memoria FIFO y su control también están presentes.

Para acceder al control y manipular la FIFO, se utiliza el AXIs como un enlace entre la FIFO y el procesador. El AXI_control actúa como un enlace adicional para permitir que el AXI_gentcount se conecte al procesador. De forma concreta, estos módulos y conexiones permiten el manejo y control de la memoria FIFO dentro del sistema.

Cuando se utiliza el AXI_count, se pueden manipular las entradas o salidas del módulo. Si es una salida, se puede escribir en el AXI_gentcount, mientras que si es una entrada, se puede leer desde allí. Al conectar los módulos AXIs al procesador, se les asigna una dirección específica, como se muestra en la tabla 3.2.1. Por ejemplo, para acceder al AXIs_control se utiliza la dirección `0x4120_0000`, y para acceder al AXIs_gentcount se utiliza la dirección `0x4200_0000`. Al escribir en una dirección, se escribirá en los 8 bits correspondientes, y también hay entradas de 18 bits. Esto se puede entender como conectar una tubería o bus al procesador para permitir el flujo de datos, y también se realiza una conexión para leer los datos que se están transmitiendo.

El módulo AXIs_control tiene dos puertos GPIO y GPIO2, que es un canal doble. GPIO2 se utiliza como entrada (lectura) cuando no se selecciona nada, pero se vuelve una salida (escritura) cuando escribimos en el GPIO. Sin embargo, estas escrituras se realizan sobre

GpioIN [5:4]	BUS ADC_FIFO	Clock FIFO	Descripción
0 0	ADC_DATA	Sys ClkOut	Traza Cascada
0 1	ADC_DATA	Sys Clk	Free Cascada
1 0	ADC_Counter	Sys ClkOut	Traza Counter
1 1	ADC_Counter	Sys Clk	Free Counter

Tabla 3.2.2: Configuración de las puertas para adquisición de datos.

el AXI_control, y el mapeo de direcciones se encuentra en 4120, con un tamaño de 64k. El mapa de direcciones es la dirección física o base utilizada para acceder a los registros y funcionalidades del módulo.

Dentro del módulo AXI_control, se dispone de un tamaño de 8 bits que nos permite seleccionar una señal de acción para realizar una determinada operación. Cada bit representa una señal de control específica, y al modificar el estado de estos bits, se elige uno de los modos de adquisición de datos disponibles. Estos modos serán descritos más adelante los cuales incluyen operaciones de iniciar la adquisición para probar diferentes funciones.

3.2.6. Modos para la Adquisición de datos

El código hecho en el framework de Python está diseñado para que, de forma sencilla se pueda hacer modificaciones respecto al modo que uno requiera usar, dichos modos son definidos como: *FreeCounter*, *TrazaCounter*, *FreeCascada* y *TrazaCascada*, donde cada modo tiene una función en específico. Estas opciones se pueden seleccionar por la configuración de las puertas lógicas encargadas de la adquisición de datos, dicha configuración se ve en la tabla 3.2.2 En las siguientes líneas se explica la función que tiene cada modo programado:

Modo FreeCounter

El modo FreeCounter, se encarga de comprobar que la adquisición de datos se da de forma continua uno a uno cada valor dentro de la memoria, esto da como resultado una representación lineal.

Modo TrazaCounter

En el modo TrazaCounter, cuando se detecta la señal y es procesada, obtiene los datos y los segmenta dado un umbral de disparo, cada segmento es una rampa con 16,32 o 64 muestras almacenadas en la memoria, debido a que contabiliza de manera continua.

Modo FreeCascada

El modo FreeCascada consiste en la adquisición libre de datos o también llamado carrera libre donde la Redpitaya toma unas muestras de potencias de 2, en concreto ya sea para 512 o 1024 datos de manera continua independientemente de los valores que estos adquieran, mientras estén en el rango de operación de la tarjeta ($\pm 1v$).

Modo TrazaCascada

El modo TrazaCascada consiste en adquirir un conjunto de 512 o 1024 datos acoplados de forma continua que superen un umbral de disparo (manual) en función de la señal a la cual se esté estudiando.

3.2.7. Arreglo experimental

El arreglo usado para probar la adquisición consistía en los siguientes instrumentos: Un generador de pulsos Aligent 81101A de 50 MHz, osciloscopio Tektronik 3020, Redpitaya STEMlab 125-14 y una laptop

Entonces, una vez realizada la programación y configuración de la tarjeta digitalizadora, se procedió a probarla adquiriendo las señales del generador de pulsos 81101A de 50 MHz. Utilizamos su salida (output) para conectarla mediante puntas a la entrada rápida (RF) de la tarjeta, tal como se muestra en la Figura 3.2.6.



Figura 3.2.6: Arreglo formado por el generador de pulsos Agilent 81101A, la tarjeta Redpitaya STEMLab 125-14 y una laptop.

Posteriormente, se llevó a cabo la prueba de los cuatro modos disponibles, siendo los más relevantes la adquisición libre de los pulsos y la adquisición por trazas de los perfiles de los pulsos cargados dentro de la tarjeta. En la Figura 3.2.8, se puede observar que se generó un pulso, el cual fue visualizado en el osciloscopio y, al mismo tiempo, se pudo ver el pulso digitalizado en la pantalla de la computadora.

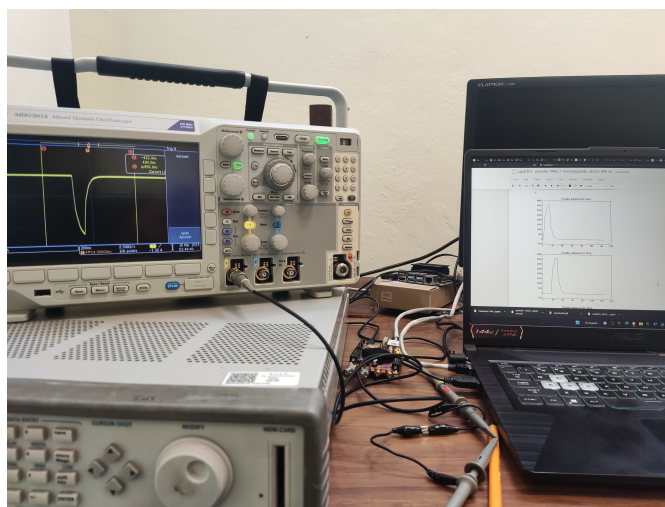


Figura 3.2.7: Pulso emitido por el generador donde podemos observar que la tarjeta puede almacenar y visualizar la misma señal.

Una vez demostrado el modo de adquisición libre, se procedió a utilizar el modo de adquisición por trazas. En la Figura 3.2.8, se

puede observar de manera análoga al caso previo que ahora es posible visualizar los perfiles de los pulsos agrupados que fueron almacenados en la memoria.

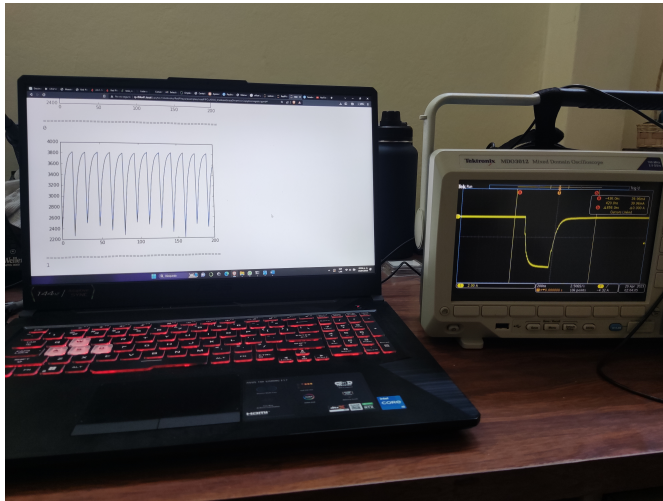


Figura 3.2.8: Captura de los perfiles de un Pulso emitido por el generador donde podemos observar que la tarjeta puede almacenar y visualizar la misma señal.

Posteriormente en la segunda parte se usaron los siguientes instrumentos: SiPM microf–J 60035 SensL, Osciloscopio Tektronik 3020, Redpitaya STEMLab 125 – 14. Fuente de voltaje para alimentar al SiPM con un voltaje de 5V y al amplificador de la señal con un voltaje de 30V. Ver Figura 3.2.11

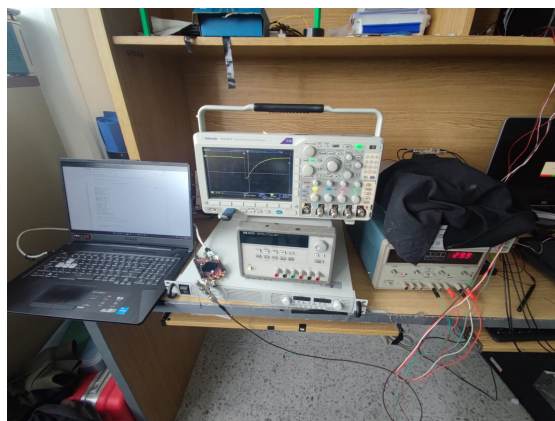


Figura 3.2.9: Arreglo experimental donde están conectados el SiPM, fuente de voltaje que lo alimenta, STEMLab 125-14 y una computadora.

la visualización de los pulsos en el osciloscopio se pueden ver e la siguiente forma, donde tenían una media de anchos de pulso de 400 ns.

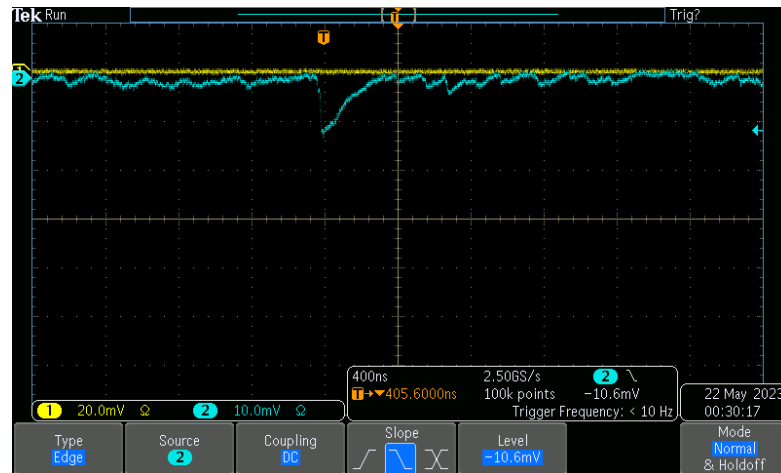


Figura 3.2.10: Pulso emitido por un SiPM visto desde un osciloscopio tektronik 3020.

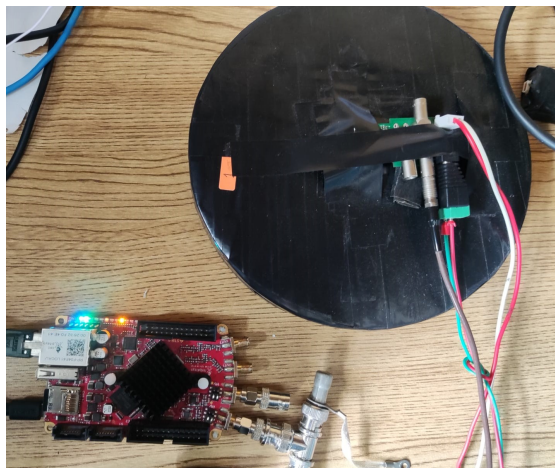


Figura 3.2.11: SiPM sin la cubierta de color negro, al lado de STEMLab 125-14.

Con este sistema de tomaron las pruebas a 16,32 y 64 muestras, lo obtenido de ellas se ve en el capítulo de resultados.

Capítulo 4

Resultados

4.0.1. Pruebas a la instrumentación

En las siguientes figuras se muestran las pruebas hechas para probar el funcionamiento de los distintos modos. Para demostrar que el sistema lograba capturar un tren de datos se ha hecho uso del modo *FreeCounter*, el cual se puede ver representado en la figura 4.0.1

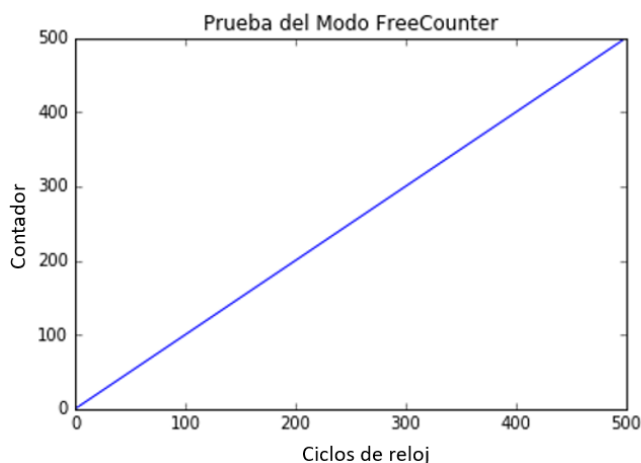


Figura 4.0.1: Comprobación del Modo FreeCounter, como resultado del conteo de muestras por STEMlab 125-14.

Posteriormente, se probó con una toma de datos en el modo *TrazaCounter* que depende de un umbral de disparo el cual nos da segmentos de recta que se componen de 16, 32 o 64 muestras, cada dato muestral se obtiene cada 8 ns lo que nos da una ventana de muestreo de al menos 128 ns, para el caso de 16 muestras, esto lo podemos ver

representado en la figura 4.0.2

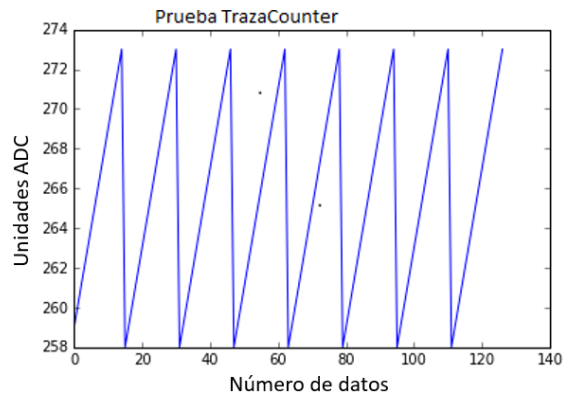


Figura 4.0.2: Comprobación del Modo de trazas del contador de 16 muestras por STEMLab 125-14.

Posteriormente, de forma análoga se probó a 32 Fig. 4.0.3

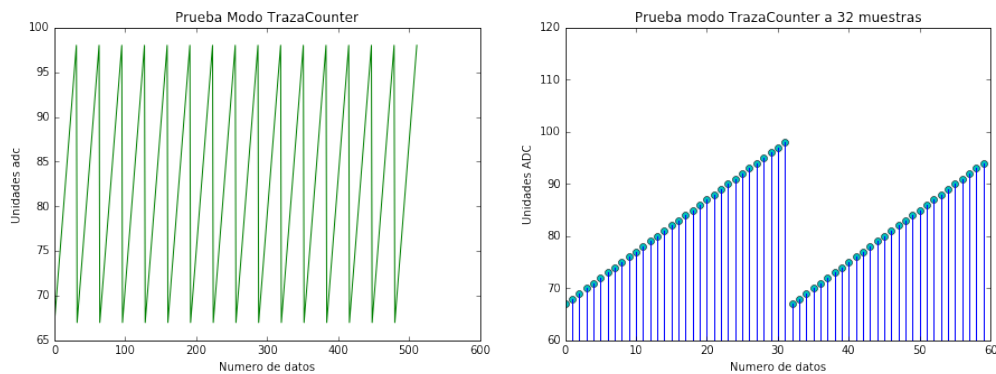


Figura 4.0.3: Comprobación del Modo de trazas del contador de 32 muestras por STEMLab 125-14.

4.0.2. Adquisición de datos con Generador de funciones tektronix AFG310 y Generador de Pulsos Aligent 81101A

Para la prueba de los modos *FreeCascade* y *Traza Cascade*, el sistema se conectó primero a un Generador de funciones tektronix AFG310 y posteriormente a un generador de Pulsos Aligent 81101A para que la tarjeta adquiriera los pulsos por medio de su entrada ADC, con el uso del generadores se tiene control sobre las características de la señal entrante, como son la amplitud, la frecuencia y el

ancho del pulso

El código utilizado en el framework de python 5.2 está diseñado para que, una vez adquiridos los pulsos emitidos por un detector, estos son de amplitud negativa, por lo tanto, se realiza un procesamiento de datos que invierta el sentido de los valores multiplicando por un número negativo, haciendo que, los pulsos adquiridos sean de amplitud positiva para una mejor comodidad en su análisis.

Se realizaron pruebas con el generador de funciones para comprobar el funcionamiento del sistema de adquisición en el modo FreeCascade. Se probaron diferentes variaciones a distintas amplitudes, incluyendo funciones triangulares, cuadradas y senoidales. Ver Figura 4.0.4

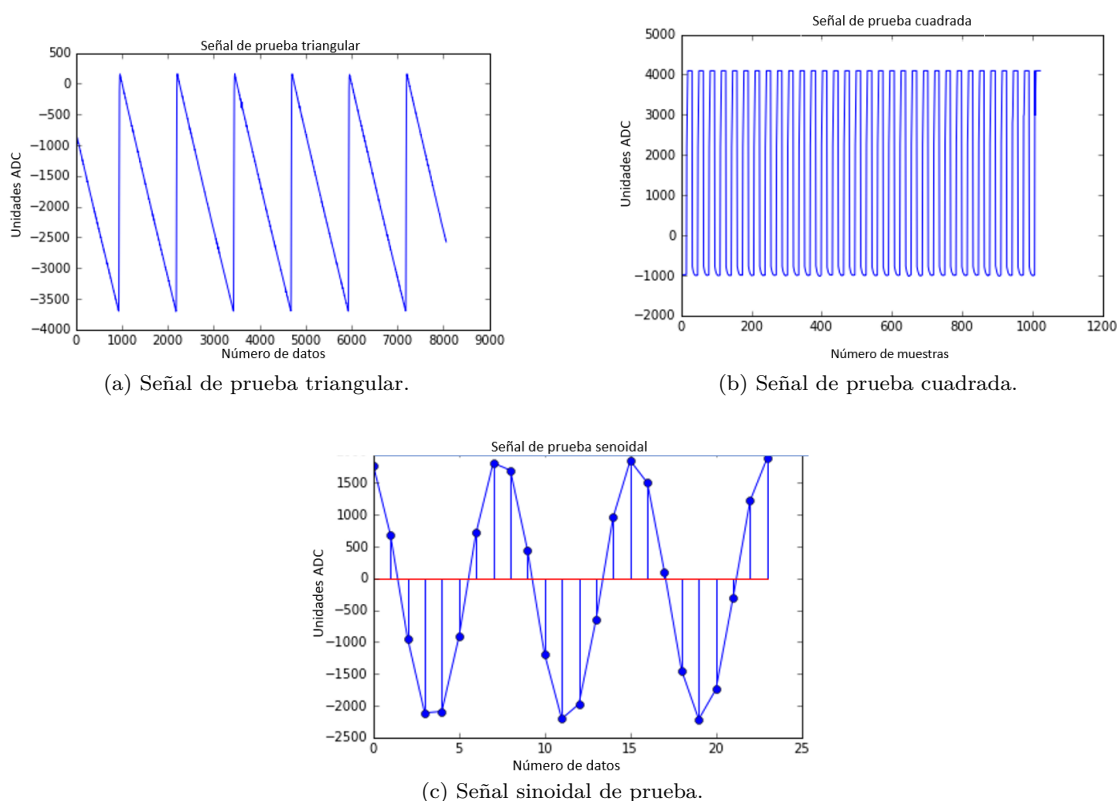


Figura 4.0.4: Visualización a diferentes señales de prueba.

Posteriormente, para probar la adquisición de muestras por el mo-

do FreeCascada se tomaron pulsos a diferentes frecuencias, el primero de ellos fue de 100 kHz, esto es a 10000 ns, como podemos ver en la figura 4.0.5

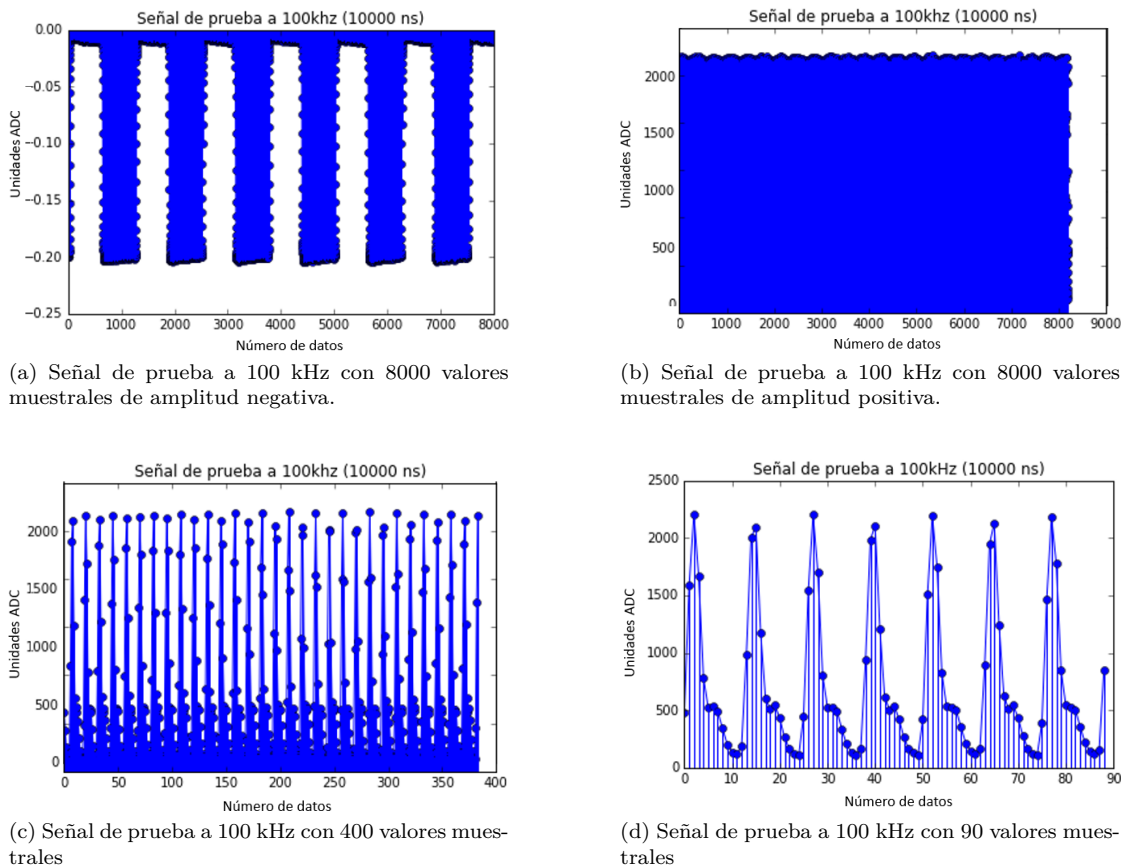
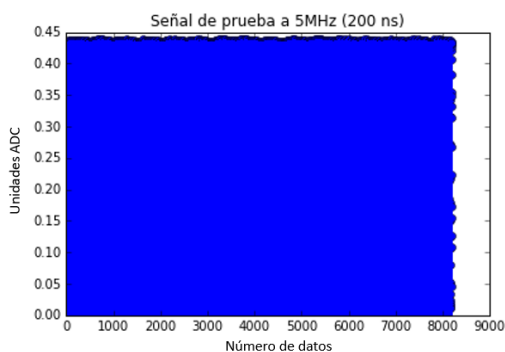
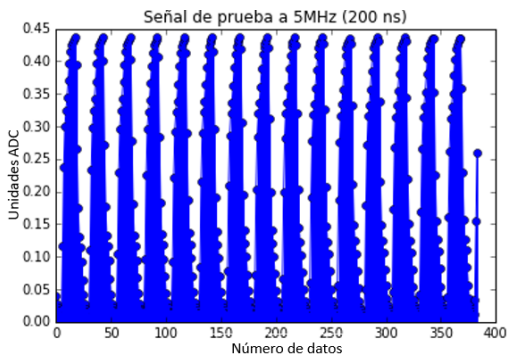


Figura 4.0.5: Visualización a diferentes tamaños de muestras de un Pulso a 100kHz.

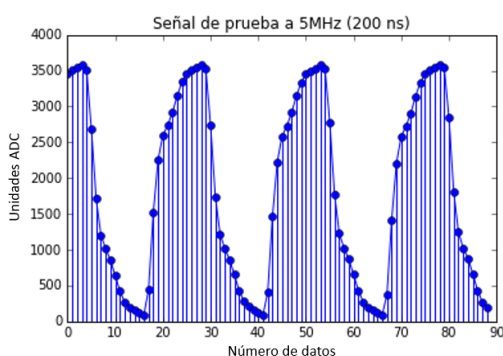
Para las siguientes pruebas se aumentó la frecuencia de prueba, el segundo ejemplo es a 5 MHz con un ancho de pulso de 200 ns a diferentes tamaños de muestras para una mejor visualización de los datos. ver figura 4.0.6



(a) Señal de prueba a 5 MHz con 8000 valores muestrales de amplitud positiva.



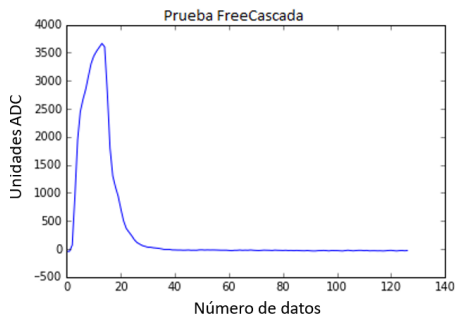
(b) Señal de prueba a 5 MHz con 400 valores muestrales de amplitud positiva.



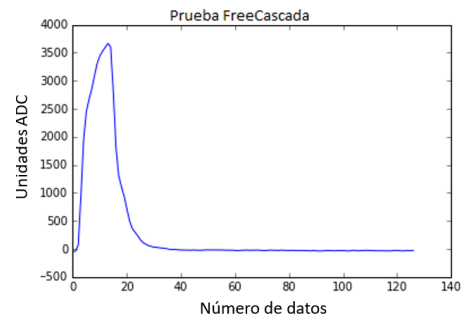
(c) Señal de prueba a 5 MHz con 90 valores muestrales.

Figura 4.0.6: Visualización a diferentes tamaños de muestras de un Pulso a 5 MHz.

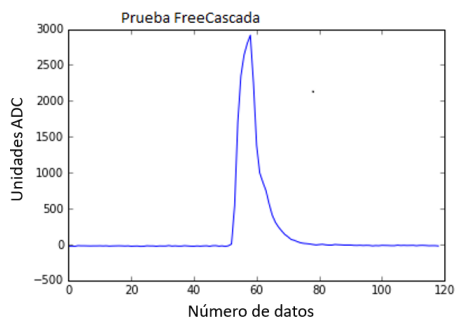
Para lograr comparar las muestras mostrados por un detector del tipo fotomultiplicador, que con las que son obtenidas por el generador de pulsos, la anchura de los pulsos debe ser similar. los parámetros del pulso fueron: Un ciclo de trabajo negativo del orden de decenas de nanosegundos y un ciclo de trabajo positivo de mayor valor. Una vez puesto a prueba que el modo *FreeCascada* podía capturar un amplio rango de frecuencias que va de los kHz a los MHz, se probaron algunos pulsos a diferentes frecuencias y con anchos de pulso variables, con una amplitud fija de -500 mV. Estos pulsos son de las frecuencias: 1 Mhz con ancho de 900 ns y 950 ns, 2 Mhz con ancho de 420 ns y 8 MHz con ancho de 100 ns.



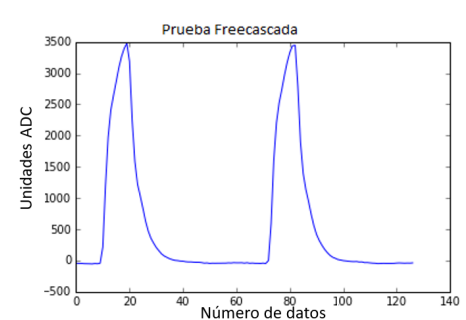
(a) Señal de prueba a 1 MHz con periodo 1μ s con ancho de pulso de 900 ns.



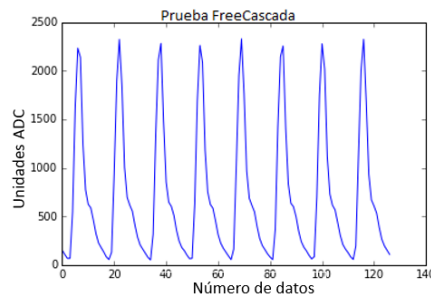
(b) Señal de prueba a 1 MHz con periodo 1μ s con ancho de pulso de 900 ns con 16 puntos muestrales.



(c) Señal de prueba a 1 MHz con periodo 1μ s con ancho de pulso de 950 ns.



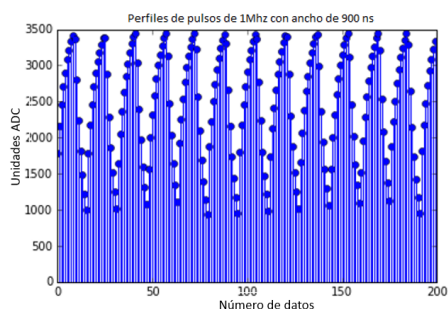
(d) Señal de prueba a 2 MHz con periodo 500 ns con ancho de pulso de 420 ns.



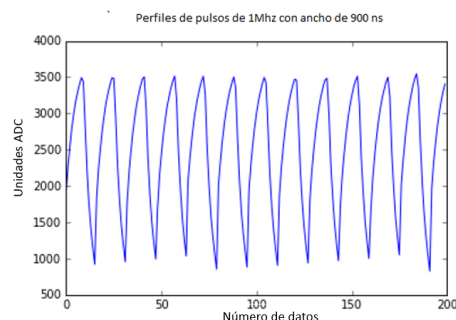
(e) Señal de prueba a 8 MHz con periodo 125 ns con ancho de pulso de 900 ns.

Figura 4.0.7: Visualización de diferentes pulsos muestreados con el modo FreeCascada.

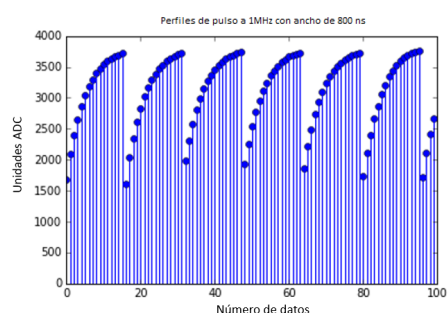
Posteriormente se probó el modo *TrazaCascada* el cual dado el umbral de disparo, donde se puede manipular de forma manual como se ve en el *script* de Python 5.1 y dado en valores hexadecimales, comienza a concatenar los perfiles de los pulsos con 16 muestras cada uno de ellos, de forma ordenada. Ver figura 4.0.8



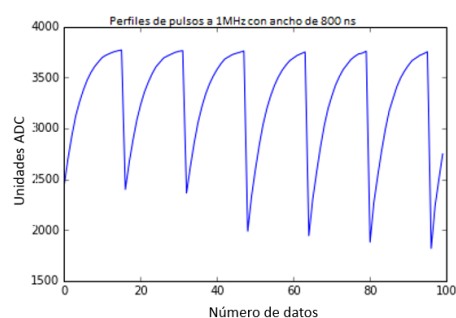
(a) Puntos de los perfiles de pulso a 1MHz y ancho de 900 ns.



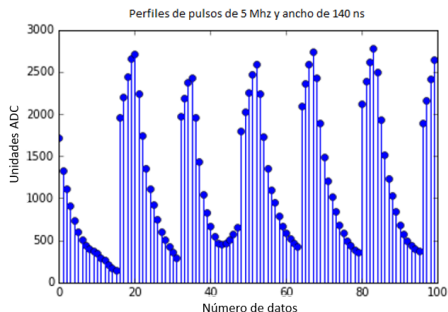
(b) Perfiles de pulso a 1MHz y ancho de 900 ns.



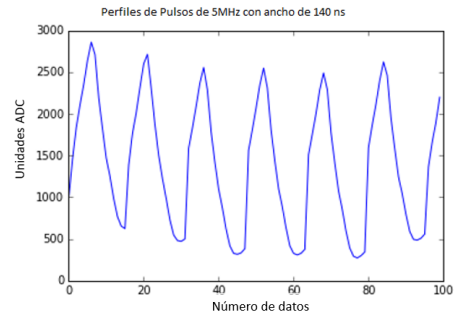
(c) Puntos de los perfiles de pulso a 1 MHz y ancho de 800 ns.



(d) perfiles de pulso a 1 MHz y ancho de 800 ns.



(e) Puntos de los perfiles de pulso a 5 MHz y ancho de 140 ns.



(f) perfiles de pulso a 5 MHz y ancho de 140 ns.

Figura 4.0.8: Visualización de diferentes perfiles de pulsos muestreados con el modo TrazaCasca.

4.0.3. Ruido generado por STEMLab 125 – 14

Mediciones de los niveles del ruido

Se cálculo la medida del ruido generado solo por la tarjeta y se ubicó la línea base con valor en unidades adimensionales de 60,82,

donde la media μ se cálculo de 2718 iteraciones de 512 muestras cada una, equivalente a 1,359,360 de muestras. En la Fig.4.0.10a se visualiza en unidades adimensionales y de manera análoga haciendo un ajuste, podemos ver el nivel de ruido en mV en la Fig. 4.0.10b el cual tiene una media $\mu = 7,485$ y desviación estándar $\sigma = 0,279$ mV.

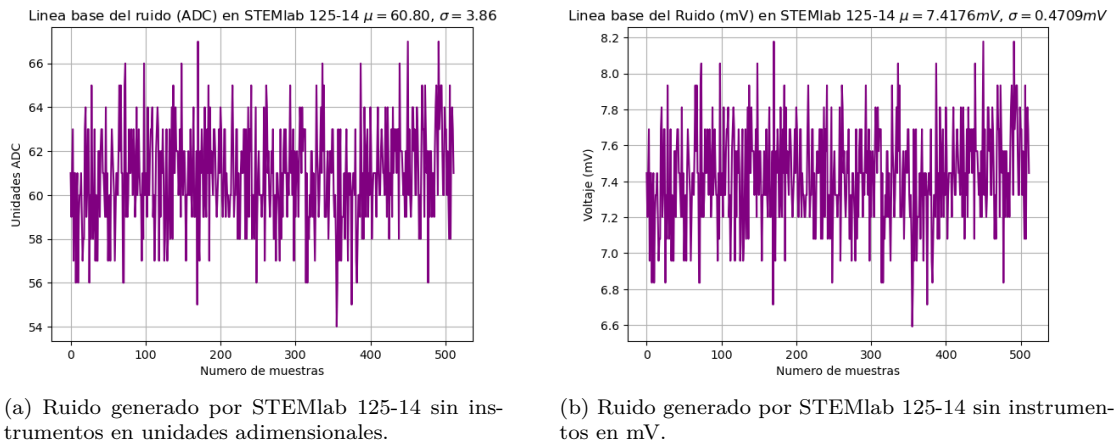


Figura 4.0.9: Niveles del ruido tanto en unidades adimensionales así como en voltaje.

Posteriormente se tomaron muestras del ruido generado con todo el sistema completo conectado que incluye el SiPM y las fuentes de voltaje de 5 V y 30 V que lo alimentan, pero con el amplificador apagado el cual es alimentado por la fuente de 30 V, donde podemos ver que el nivel de ruido se mantiene en niveles similares, esto lo podemos observar al calcular la media de 3250 iteraciones de 512 muestras, lo equivalente a 1,664,327 datos muestrales y obtenemos una media total $\mu = 7,20$ mV, y con desviación estándar $\sigma = 0,35$ mV, también se han representado las muestras de ambos casos mediante los siguientes histogramas de ruido.

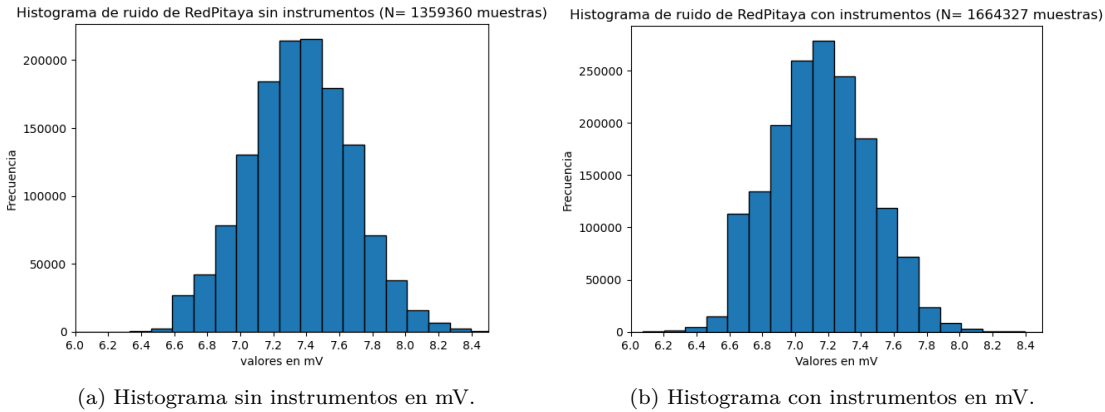


Figura 4.0.10: Histogramas de las distribuciones del ruido donde STEMLab 125-14 esta con y sin instrumentos extras visualizado en voltaje (mV).

4.0.4. Perfiles de trazas a 16 muestras de un SiPM

Las siguientes pruebas se hicieron usando un SiPM microf-J 60035 SensL, en las primeras pruebas se realizaron a 16 muestras la cual generaba una ventana temporal de 128ns, con un buffer de 1000 muestras por cada toma de datos, tomando en cuenta que se encuentra en unidades adimensionales. Sin embargo, el tamaño de los pulsos resultaron de mayor anchura por lo que, se obtuvo información incompleta de ellos, al concatenar los perfiles, como podemos ver en la Fig. 4.0.11.

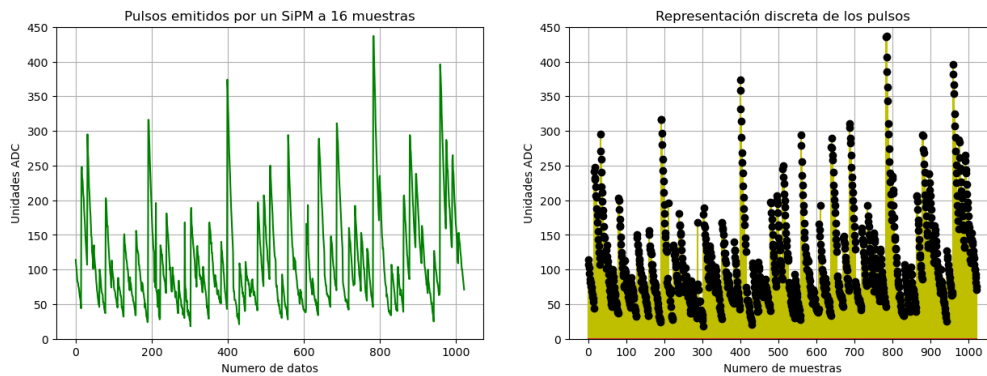


Figura 4.0.11: Perfiles de pulsos detectados por un SiPM así como la visualización de las 16 muestras de cada uno en unidades adimensionales.

Así mismo podemos ver en la Figura Fig.4.0.12 un acercamiento

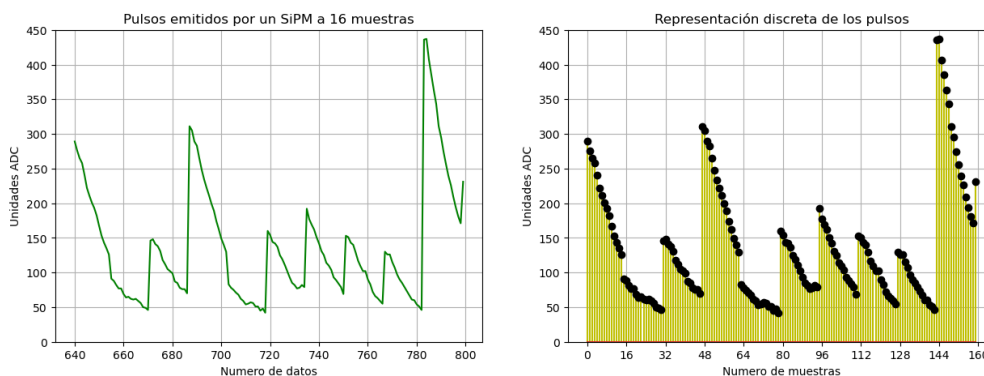


Figura 4.0.12: Visualización sobre algunos de los perfiles de pulsos detectados por un SiPM en el rango de 640 a 800 dentro del almacenamiento de 1024 muestras.

de los perfiles capturados en particular de trata del segmento entre los puntos 640 y 800 almacenados en la memoria.

4.0.5. Perfiles de trazas a 32 muestras de un SiPM

Visto en la Figura anterior, se puede apreciar que la configuración inicial de la adquisición de datos no permitía observar completamente las señales emitidas por el SiPM. Por lo tanto, se realizó una reconfiguración con el objetivo de obtener la mayor información posible. Además, se modificó el tren de datos a 512 muestras para realizar una adquisición de perfiles en un menor tiempo. Es importante destacar que esta configuración puede ser ajustada según las necesidades, ya sea aumentando o disminuyendo el tren de datos en potencias de 2.

Adicionalmente, se duplicó la ventana temporal de 16 muestras a 32 muestras. En dicha ventana temporal de 256 ns, se pudo observar que los pulsos con una menor amplitud estaban siendo adquiridos en su totalidad, alcanzando el nivel de la línea base de ruido. Sin embargo, los pulsos de mayor amplitud no fueron detectados en su totalidad. Por lo que se reconfiguro nuevamente la adquisición. Ver Figura4.0.13.

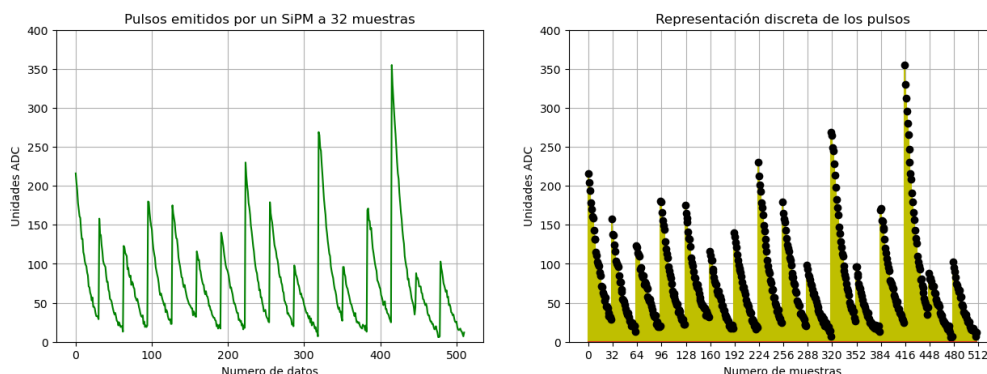


Figura 4.0.13: Perfiles de pulsos detectados por un SiPM así como la visualización de las 32 muestras de cada perfil en unidades adimensionales.

Análogamente al caso de 16 muestras, se hace un acercamiento de los perfiles capturados en las posiciones en el rango de 300 a 500 dentro del almacenamiento de 512 muestras.

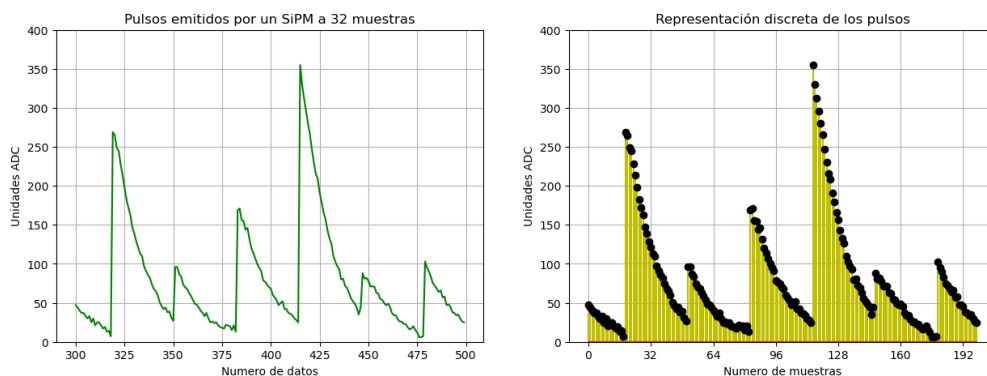


Figura 4.0.14: Visualización sobre algunos de los perfiles de pulsos detectados por un SiPM en el rango de 300 a 500 dentro del almacenamiento de 512 muestras.

4.0.6. Perfiles de trazas a 64 muestras de un SiPM

Dado que los pulsos de amplitud mayor a 250 unidades ADC, no eran procesados completamente, se volvió a reconfigurar la adquisición al doble, esto es, subir la ventana temporal a 64 muestras siendo equivalente a 512 ns, esto es debido a que el promedio de los pulsos emitidos por el SiPM, se encuentra alrededor de 400 ns de anchura como se puede apreciar en la Figura 3.2.10. Finalmente, con dichos

cambios se obtuvieron pulsos completos a pesar de su amplitud. Ver Figura 4.0.16

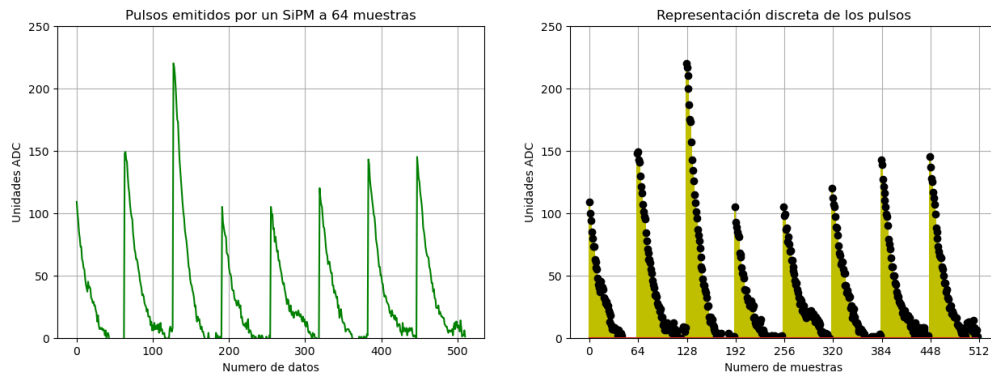


Figura 4.0.15: Perfiles de pulsos detectados del SiPM a 64 muestras.

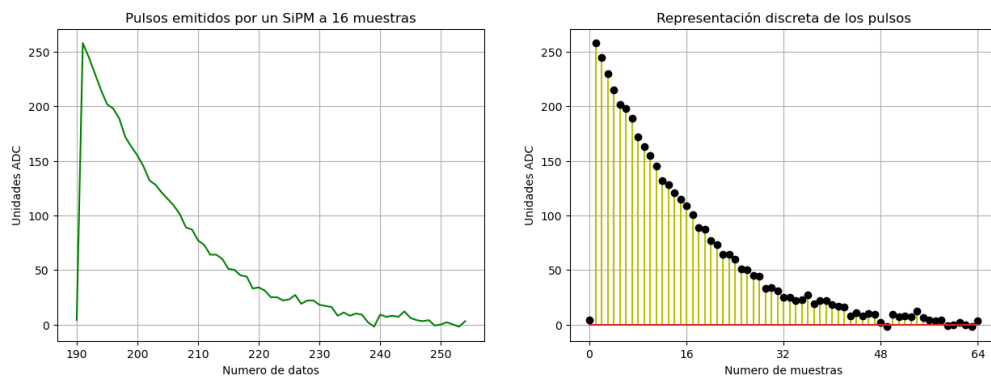
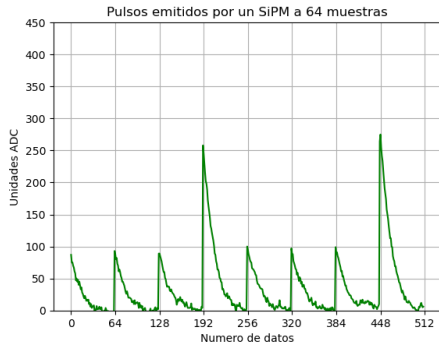
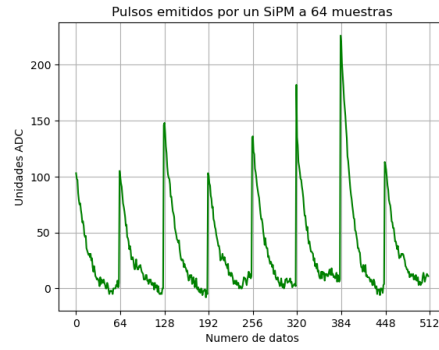


Figura 4.0.16: Visualización sobre algunos de los perfiles de pulsos detectados por un SiPM en el rango de 190 a 255 dentro del almacenamiento de 512 muestras.

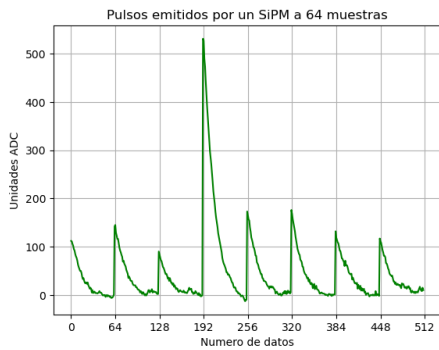
En las siguientes figuras podemos ver diferentes toma de datos de los perfiles de pulsos con diferentes amplitudes adquiridos dado un umbral de disparo por el modo de adquisición de *TrazaCascada*.



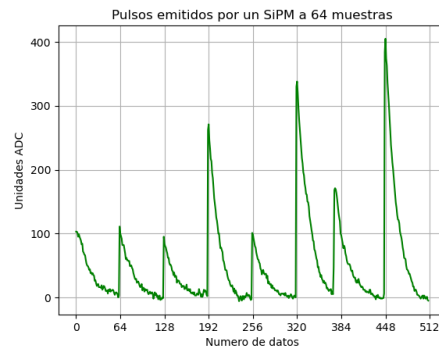
(a) Primer ejemplo de diferentes perfiles de pulsos almacenados en la memoria.



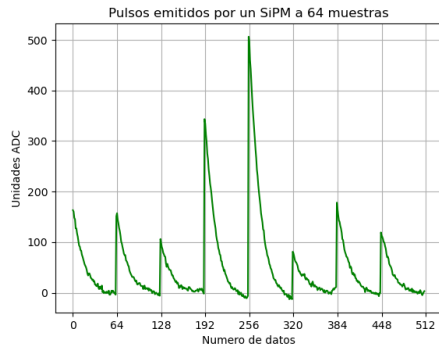
(b) Segundo ejemplo de diferentes perfiles de pulsos almacenados en la memoria.



(c) Tercer ejemplo de diferentes perfiles de pulsos almacenados en la memoria.



(d) Cuarto ejemplo de diferentes perfiles de pulsos almacenados en la memoria.



(e) Quinto ejemplo de diferentes perfiles de pulsos almacenados en la memoria.

Figura 4.0.17: Diferentes conjuntos de perfiles de pulsos adquiridos por la Redpitaya.

A partir de la toma de datos de los perfiles se obtiene la amplitud de los perfiles de una base de datos generada durante una toma

constante en un tiempo de 70 horas, lo que resultó en un total de 2,857 millones de datos muestrales, equivalentes a 44,647 pulsos. El siguiente histograma muestra la frecuencia de las amplitudes de los pulsos detectados.

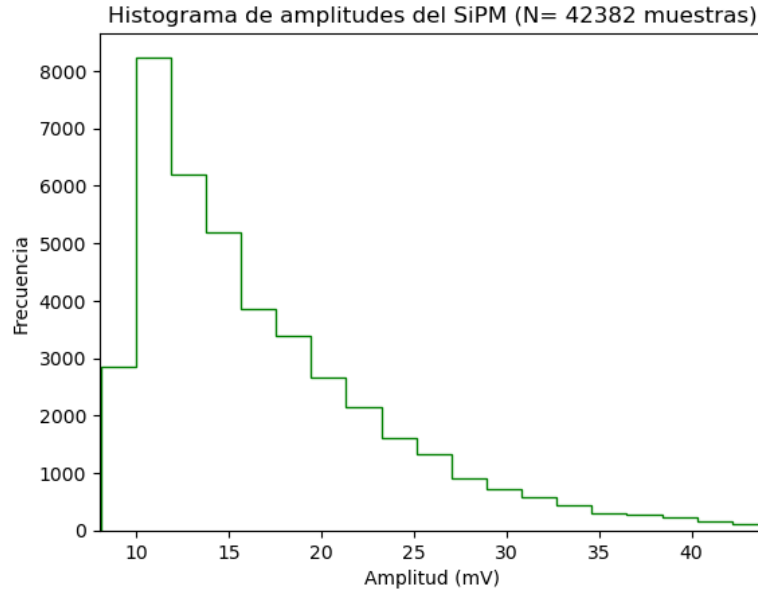


Figura 4.0.18: Histograma de amplitud de los perfiles almacenados.

Para obtener el histograma de carga se integra el voltaje, donde dada la relación de corriente, la carga es $I = \frac{dq}{dt}$ donde I es la corriente, q la carga y t el tiempo. De la ley de Ohm $V = IR$, se tiene:

$$\frac{dq}{dt} = \frac{V}{R}$$

$$Q_t = \frac{1}{R} \int V dt$$

De forma particular para el SiPM

$$Q_t = \frac{1}{7R_s} \int_{t_0}^{t_1} V dt$$

De forma discreta

$$Q_t = \frac{1}{7R_s} \Sigma(V * \Delta t)$$

Donde Δt corresponde a 8 ns sumando un total de 512 ns (el tiempo de muestreo por cada perfil), V el voltaje y R_s la impedancia de 50Ω . En la Figura 4.0.19 podemos ver el histograma de carga correspondiente.

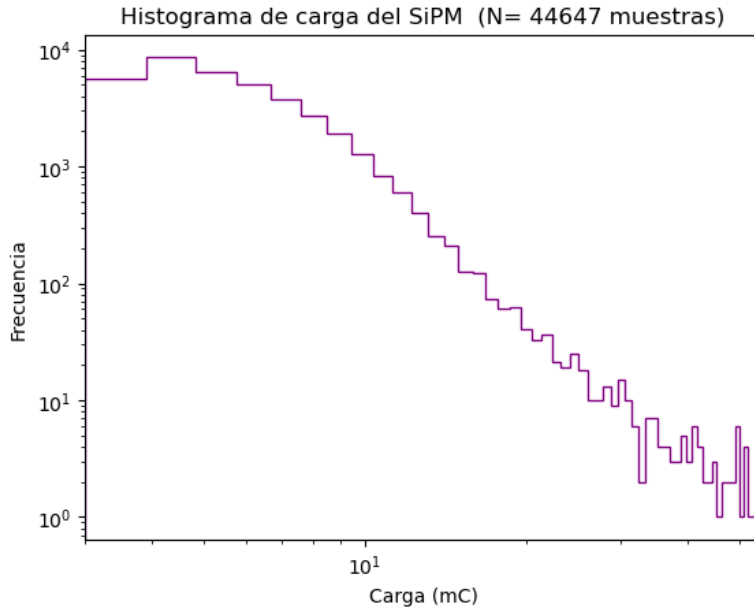


Figura 4.0.19: Histograma de carga.

En la figura. 4.0.19, los valores con menor carga corresponden a eventos de partículas únicas. Sin embargo, a medida que aumenta la energía, la frecuencia de eventos disminuye, lo que indica que se trata de eventos producidos por chubascos atmosféricos.

Capítulo 5

Conclusiones y trabajo a futuro

Se ha desarrollado e implementado un sistema y códigos de control que permiten interactuar con la tarjeta Redpitaya, con esto se adapta funcionalidad de la Redpitaya como un instrumento versátil y reprogramable debido a su naturaleza de ser basada en un *System on Chip*. Esto permite actualizaciones tanto en el hardware (*Logic Programming*) como en el software *Software Programming*.

Se ha demostrado que al utilizar programación en un lenguaje de descripción de hardware de alto nivel, es posible configurar el dispositivo para adquirir y procesar datos, con el fin de realizar análisis y visualización posterior de los mismos. Es importante tener en cuenta que este enfoque es efectivo siempre y cuando el ancho de los pulsos sea mayor al límite de la frecuencia de muestreo, que en este caso es de 8 ns.

5.0.1. Perspectivas futuras:

- Se pretende dar continuidad a la instrumentación de rayos cósmicos, en particular, mejorando el sistema de adquisición, la conexión a distancia y comparándolo con otros instrumentos disponibles.
- Extender la arquitectura a dos canales de entrada.
- Se planea implementarlo a corto plazo en un WCD (Detector Cherenkov de Agua) y realizar la toma de datos.

Apéndice

5.1. Enlace de conexión entre una tarjeta Redpitaya y una PC

Para establecer una conexión de enlace entre Redpitaya STEMLab 125 – 14 y una PC, se deben seguir una serie de instrucciones específicas. Dependiendo de las necesidades del usuario, existen dos formas de realizar esta conexión. Es posible crear un enlace desde una sesión en el sistema operativo Windows o utilizar una variante del sistema operativo basado en Linux, lo cual requiere tener en cuenta algunos detalles adicionales.

El Linux, lo primero es crear una conexión local entre la tarjeta STEMLab 125 – 14 y la computadora, esto se hace, conectando físicamente la tarjeta al puerto RJ45 y un puerto USB para alimentarla a 5V. Se configura la conexión en la computadora, ingresando al centro de redes y creando un segundo perfil en la red local para poder configurar la red cableada. en este perfil se configuran las propiedades al modificar el protocolo de internet haciendo que “IPv4” y “IPv6” estén en la opción de “solo enlace local”.

En Windows, se conecta STEMLab 125 – 14 a la red. Esto se logra usando el puerto Ethernet RJ45 de la tarjeta y se conecta directamente a una entrada disponible de red en el Router. De igual manera se conecta a la computadora por medio de un cable Ethernet hacia el mismo Router donde está conectada la tarjeta. Otra opción disponible y la cual se ha implementado, consiste en conectar de forma

local (o directa) la Redpitaya STEMLab 125 – 14 al puerto RJ45 de la computadora de trabajo. Esta conexión se muestra en la figura 5.1.1.

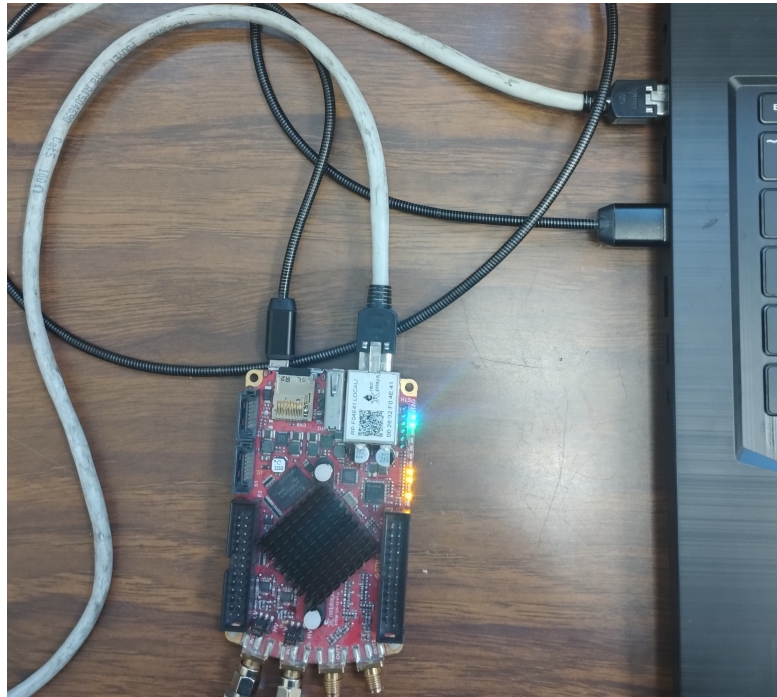


Figura 5.1.1: vista de la conexión local.

Posteriormente, se abre una *terminal* en Linux o una terminal de *powershell* en Windows y se ejecuta el comando: `ssh root@IP`. De forma particular, como la conexión es local, es suficiente con escribir la dirección local asociada a la tarjeta con el nombre de `rp-f04e41.local`, este etiqueta es específica con cada tarjeta y su ubicación es mostrada en la figura 5.1.2.



Figura 5.1.2: Dirección particular de la tarjeta.

La clave de acceso a la sesión remota es por defecto *root*, en caso de cambiar y olvidar la clave acceso es complicado recuperar la clave por lo que se recomienda guardarla o no poner claves de acceso difíciles de recordar. Respecto a los comandos de uso y navegación son los mismos que un entorno de ubuntu de linux. En la figura 5.1.3 se visualiza la interfaz de conexión

```

Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS C:\Users\randk> ssh root@rp-f04e41.local
root@rp-f04e41.local's password:
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.9.0-xilinx armv7l)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage
#####
# Red Pitaya GNU/Linux Ecosystem
# Version: 0.98
# Build: 696
# Branch:
# Commit: 38ef64d5771e52f1c1ee6c79ebd52a72c81de8be
# U-Boot: "redpitaya-v2016.4"
# Linux Kernel: "branch-redpitaya-v2017.2"
# Pro Applications: ""
#####
Last login: Wed Feb 22 04:46:27 2023 from 169.254.288.153
root@rp-f04e41:~#
    
```

Figura 5.1.3: Interfaz en *powershell* para acceder al entorno de la tarjeta.

Para establecer la conexión entre el equipo de cómputo y la tarjeta, es necesario ingresar en la barra de direcciones de cualquier navegador la siguiente dirección: `rp-f04e41.local`. Como se mencionó previamente en la figura 5.1.2, los datos de acceso se encuentran impresos en la tarjeta, por lo que el número de serie es específica cada una de ellas. La interfaz de la página se muestra en la figura 5.1.4.

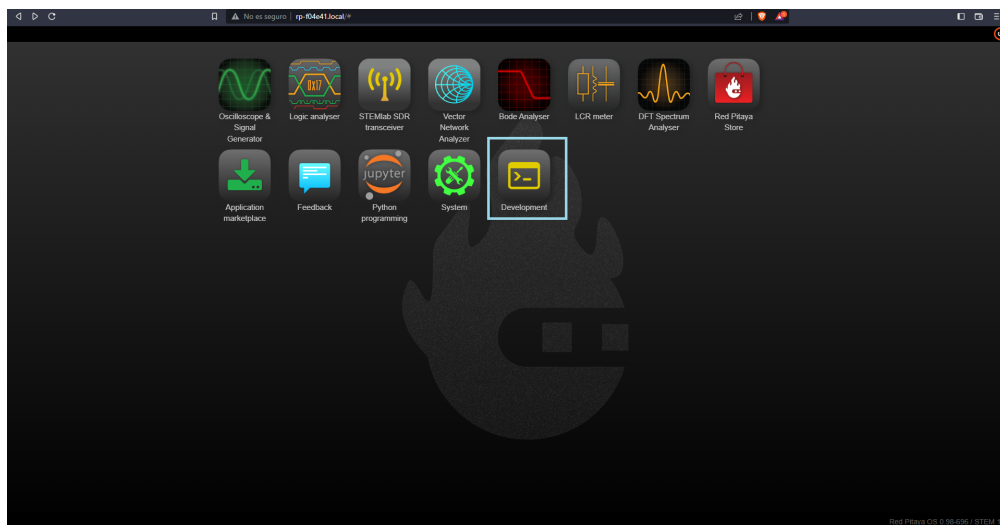


Figura 5.1.4: Interfaz Web de Redpitaya.

En la interfaz Web, se debe seleccionar la opción de *Development* y luego elegir *SCPI Server* de las opciones disponibles, como se muestra en la figura 5.1.5. Una vez dentro de esta sección, se realiza la conexión seleccionando *RUN*, tal como se muestra en la figura 5.1.6. Al completar este paso, se obtendrá la dirección IP asociada a la tarjeta utilizada y el puerto de conexión correspondiente. Esto indica que la conexión se ha establecido. Luego, se debe regresar al menú anterior y seleccionar la opción *Python programming* para ejecutar los scripts necesarios, como se observa en la figura 5.1.7.

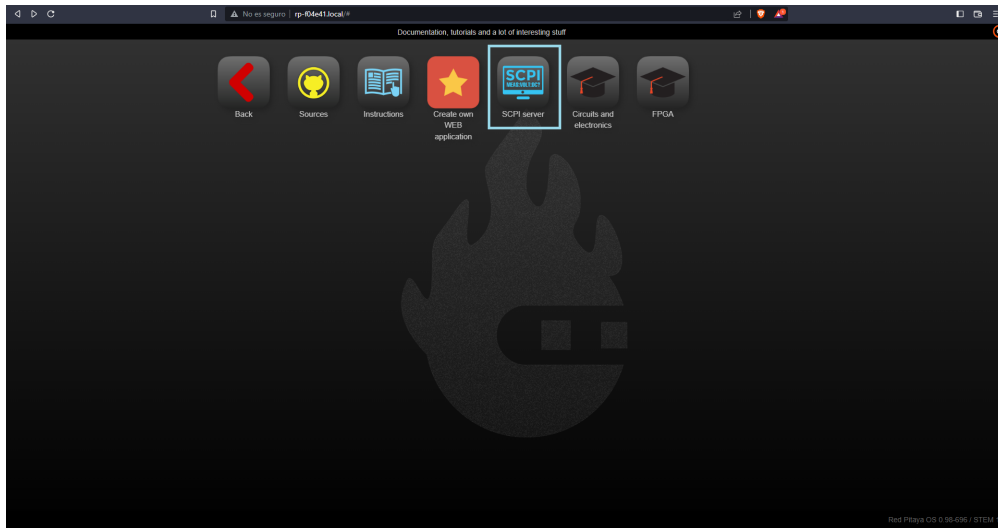


Figura 5.1.5: Menú de *Development*.

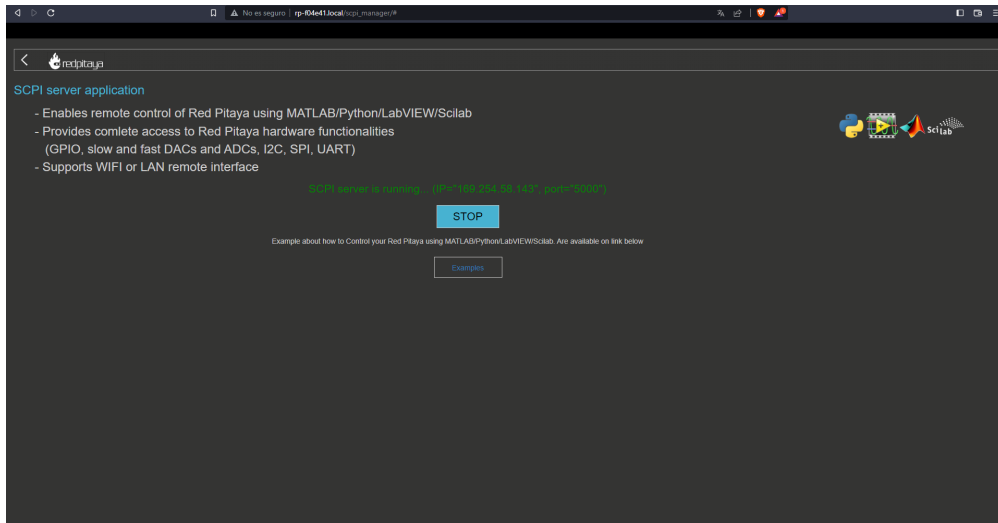


Figura 5.1.6: Conexión establecida con la tarjeta.

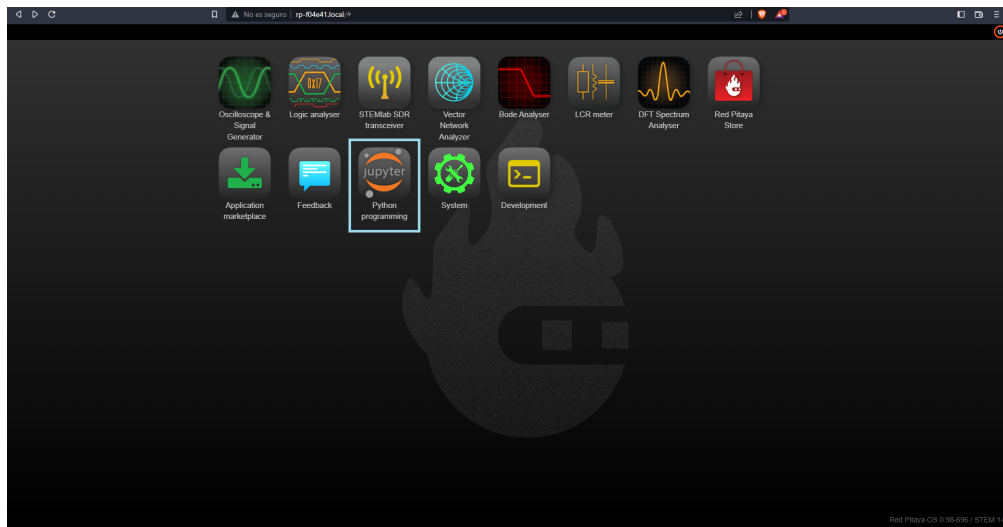


Figura 5.1.7: Selección de la opción *Python programming* para manipular el control del *firmware* de la adquisición de datos.

Ahora, regresando a la interfaz de *Powershell*, nos movemos por el directorio hasta llegar al punto donde se encuentren los programas que se encargan de la configuración de los módulos de la FPGA, en particular se trata de la siguiente dirección: `root@rp-f04e41.local/Redpitaya`.

Es importante tener cargado el archivo *bitstream* llamado `readfifo_linebase.bit` y correr el comando siguiente `cat readfifo_linebase.bit > /dev/xdevcfg`, de esta forma se está listo para ejecutar el código que manipula la parte lógica del SoC desde el ARM utilizando python.

Apéndice 2

5.1.1. Scripts en python

Listing 5.1: Código encargado de ajustar el umbral de disparo.

```
1 from periphery import MMIO
2 import numpy as np
3 ADDRESS_RANGE = 0x1000
4 ADDRESS_OFFSET = 0x10
5 regset = MMIO(0x41210000, 0xC)
6 regset.write32(0,0x96) # <— Valor de umbral, disparo
7 resolucion = 0.1220703125 #Escala de voltaje es de +-1 v
8 voltaje = int(regset.read32(0x0))*resolucion
9 print('en unidades adimensionales tenemos el valor de ')
10 print(regset.read32(0x0))
11 print('En escala de voltaje el valor correspondiente es ')
12 + str(voltaje) + str(' mv')
```

Listing 5.2: Segmento encargado de la lectura de los datos en el framework de Python.

```
1 import time
2 import matplotlib.pyplot as plt
3 import pandas as pd
4 %matplotlib inline
5 import numpy as np
6
7 from periphery import MMIO
8 import numpy as np
9 ADDRESS_RANGE = 0x1000
10 ADDRESS_OFFSET = 0x10
11
```

```

12 TrazaCascada = 0x00
13 FreeCascada = 0x10
14 TrazaCounter = 0x20
15 FreeCounter = 0x30
16
17 #Selección del modo de adquisición
18 modo = TrazaCounter
19
20 regset = MMIO(0x41200000, 0xc)
21 regset.write32(0, modo | 0x08) #
22
23 #definimos el nombre del archivo
24 nombre_archivo = "toma_32muestra.csv"
25
26 #crear un dataframe vacío para almacenar los datos
27 df = pd.DataFrame()
28
29 def twos_comp(val, bits):
30     """compute the 2's complement of int value val"""
31     # if sign bit is set e.g., 8bit: 128-255
32     if (val & (1 << (bits - 1))) != 0:
33         # compute negative value
34         val = val - (1 << bits)
35     return val
36
37 regset = MMIO(0x41200000, 0xc)
38
39 datos = []
40 for i in range(5):
41     datos = []
42     # nrst = 1; cntrl_clk = 0; rd_en_out = 0; wr_en_out = 0;
43     regset.write32(0, modo | 0x08)
44
45     # nrst = 0; cntrl_clk = 0; rd_en_out = 0; wr_en_out = 1;
46     regset.write32(0, modo | 0x01)
47
48     # en mascara los 14 bits menos significativos
49     dato = 0x10000 & regset.read32(0x08)
50     while(dato != 0x10000 ):
51         # en mascara los 14 bits menos significativos
52         dato = 0x10000 & regset.read32(0x08)
53         # nrst = 0; cntrl_clk = 0; rd_en_out = 0; wr_en_out = 1;
54         regset.write32(0, modo | 0x01)
55
56     tmp = regset.read32(0x08)
57     dato = 0x20000 & tmp # en mascara los 14 bits menos significativos
58     intDatos = []
59     while(dato != 0x20000 ):
60         # nrst = 0; cntrl_clk = 0; rd_en_out = 1; wr_en_out = 0;

```

```
61     regset.write32(0, modo | 0x02)
62     # nrst = 0; cntrl_clk = 1; rd_en_out = 1; wr_en_out = 0;
63     regset.write32(0, modo | 0x06)
64     datos.append(tmp)
65     tmp = regset.read32(0x08)
66     # en mascara los 14 bits menos significativos
67     dato = 0x20000 & tmp
68     # en mascara los 14 bits menos significativos
69     tmp = 0x3FFF & tmp
70     valor=twos_comp(tmp,14)
71     intDatos.append(valor)
72
73     #almacenar los datos de Redpitaya en un archivo
74     #csv para lecturas externas en Redpitaya
75     df_datos =pd.DataFrame(intDatos)
76     df_datos.to_csv('datos_32muestrasprueba'+str(i) +
77     '.csv', header=False, index=False)
```


Capítulo 6

*Apendice 3

6.0.1. Principales *scripts* en VHDL

Listing 6.1: Código en VHDL de CONTROL.

```
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3 use IEEE.STD_LOGIC_UNSIGNED.ALL;
4
5 entity control is
6     Port ( sys_clk      : in STD_LOGIC;
7           --cmpAB      : in std_logic;
8           gpioIN       : in std_logic_vector(7 downto 0);
9           gpioCount    : in std_logic_vector(15 downto 0);
10          umbral        : in std_logic_vector(13 downto 0);
11          ADC_DATA      : in std_logic_vector(15 downto 0);
12          --ADC_counter: in std_logic_vector(27 downto 0);
13          ADC_FIFO      : out std_logic_vector(15 downto 0);
14          gpioOut       : out std_logic_vector(17 downto 0);
15          leds          : out std_logic_vector(7 downto 0);
16          --cntrl_clk   : in STD_LOGIC;
17          --wr_en, rd_en : in STD_LOGIC;
18          wr_en_out, rd_en_out : out STD_LOGIC;
19          full, empty   : in std_logic;
20          --full_out, empty_out : out std_logic;
21          srst         : in std_logic;
22          fifo_clk      : out STD_LOGIC;
23          nrst         : out std_logic
24          );
25 end control;
26
27 architecture control_arch of control is
```

```

28 signal cntrl_clk : STD_LOGIC;
29 signal sys_traza : std_logic;
30 signal sys_clkOut : std_logic;
31 signal sys_clkOutTmp : std_logic;
32
33
34 signal cmpAB : std_logic;
35 signal ADC_DATA_short : std_logic_vector(13 downto 0);
36 component cntrlumbral is
37     Port ( clk : in STD_LOGIC;
38           rst : in STD_LOGIC;
39           cmpAB : in STD_LOGIC;
40           clkRate : out std_logic;
41           rstQ : out std_logic;
42           sys_clk : out STD_LOGIC);
43 end component cntrlumbral;
44 signal ADC_counter : std_logic_vector(27 downto 0);
45 component complemento_a_dos is
46     Port ( input_bus : in std_logic_vector(13 downto 0);
47           output_bus : out std_logic_vector(13 downto 0));
48 end component complemento_a_dos;
49 signal ADC_counte_complementtwo: std_logic_vector(15 downto 0);
50
51 begin
52 with gpioIN(5 downto 4) select
53
54     ADC_counte_complementtwo <= ADC_DATA
55                                     when "00" | "01",
56                                     -- ADC Counter
57                                     ADC_counter(15 downto 0) when others;
58
59 uComplemento_a_dos: complemento_a_dos port map (
60     input_bus => ADC_counte_complementtwo(13 downto 0),
61     output_bus => ADC_DATA_short
62 );
63 process(sys_clk , gpioIN(3))
64 begin
65     if gpioIN(3) = '1' then
66         ADC_counter <= (others => '0');
67     else
68         if rising_edge(sys_clk) then
69             ADC_counter <= ADC_counter + '1';
70         end if;
71     end if;
72 end process;
73
74 process(sys_clk)
75 begin
76     if rising_edge(sys_clk) then
77         if ADC_DATA_short(13 downto 0) < umbral then

```

```

77         cmpAB <= '1';
78     else
79         cmpAB <= '0';
80     end if;
81 end if;
82 end process;
83 —
84 wr_en_out <= gpioIN(0);
85 rd_en_out <= gpioIN(1);
86 cntrl_clk <= gpioIN(2);
87 nrst      <= gpioIN(3);
88 —        <= gpioIN(4);
89 —        <= gpioIN(5);
90 —        <= gpioIN(6);
91 —        <= gpioIN(7);
92 with gpioIN(5 downto 4) select
93     ADC_FIFO <= ADC_DATA           when "00" | "01", —ADC la cascada
94             ADC_counter(15 downto 0) when others;    — ADC Counter
95
96 with gpioIN(5 downto 4) select
97     sys_clkOutTmp <= sys_clkOut when "00" | "10", — clock en modo traza
98             NOT sys_clk when others;    — clock en modo free
99 with gpioIN(0) select
100     fifo_clk <= sys_clkOutTmp when '1', — depende de traza o free
101             cntrl_clk when others;    — Lectura en python
102
103 uCntrlumbral: cntrlumbral port map(
104     clk => sys_clk, —: in STD_LOGIC;
105     rst => gpioIN(3), — : in STD_LOGIC;
106     cmpAB => cmpAB, — : in STD_LOGIC;
107     clkRate => open, — : out std_logic;
108     sys_clk => sys_clkOut — : out STD_LOGIC
109 );
110 —
111 gpioOut <= empty & full & gpioCount; — gpioOut es de 18 bits
112 ————— B7 % B6 B5 B4
113 leds <= ADC_counter(24 downto 23) & gpioIN(3) & gpioIN(2)
114 —B3 B2 B1 B0
115 & empty & full & gpioIN(1) & gpioIN(0);
116 — gpioIN(0)
117 end control_arch;

```

 Listing 6.2: *script* encargado del Umbral

```

1
2 library IEEE;
3 use IEEE.STD_LOGIC_1164.ALL;
4 use IEEE.STD_LOGIC_ARITH.ALL;
5 use IEEE.STD_LOGIC_UNSIGNED.ALL;
6
7 entity cntrlumbral is
8     Port ( clk : in  STD_LOGIC;
9           rst : in  STD_LOGIC;
10          cmpAB : in  STD_LOGIC;
11          clkRate : out std_logic;
12          sys_clk : out STD_LOGIC);
13 end cntrlumbral;
14
15 architecture cntrlumbral_arch of cntrlumbral is
16 type state is (s0 ,s1 ,s2 ,s3 ,s4 );
17 signal currState ,nextState : state;
18 signal nextCounter ,currCounter : std_logic_vector(4 downto 0);
19 signal nextClkRate ,currClkRate: std_logic;
20 begin
21     process (clk , currState , currCounter , cmpAB , currClkRate)
22     begin
23         nextState <= currState;
24         nextCounter <= currCounter;
25         sys_clk <= '0';
26         nextClkRate <= currClkRate;
27         case currState is
28             when s0 =>
29                 nextClkRate <= '0';
30                 nextCounter <= (others => '0');
31                 nextState <= s1;
32
33             -- un punto a superado el umbral
34             when s1 =>
35
36             -- adc < umbral , TRUE. ADC no ha superadoumbral
37                 if cmpAB = '1' then
38                     nextState <= s1;
39                 else
40
41             -- -- adc < umbral , FALSE. ADC YA hasuperado umbral
42                 nextState <= s2;
43             end if;
44
45             -- un segundo punto consecutivo a superado el umbral
46             when s2 =>
47                 if cmpAB = '1' then
48                     nextState <= s1;
49                 else

```

```

50         nextState    <= s3;— 0 para seguir
51         end if;
52     — se generan 16 puntos para la memoria FIFO
53     when s3 =>
54         nextClkRate <= '1';
55         sys_clk <= clk;
56         nextCounter <= currCounter + '1';
57     —if currCounter < "01110" then — 16 muestras
58     —if currCounter < "11110" then — 32 muestras
59     — 64 muestras
60         if currCounter < "111100" then
61             nextState <= s3;
62         else
63             nextState <= s4;
64         end if;
65     when s4 =>
66         nextClkRate <= '0';
67     — adc < umbral, TRUE. La se al ya bajo, ADC no ha superado umbral
68         if cmpAB = '1' then
69             nextState <= s0;
70         else
71     —adc < umbral, TRUE. La se al sigue arriba, ADC no ha superado umbral
72             nextState    <= s4;
73         end if;
74     when others =>
75         nextState    <=s0;
76     end case;
77 end process;
78 process(clk ,rst)
79 begin
80     if rst = '1' then
81         currState <= s0;
82         currCounter <= (others =>'0');
83     else
84         if rising_edge(clk) then
85             currCounter <= nextCounter;
86             currState <= nextState;
87             currClkRate <=nextClkRate;
88             clkRate <= currClkRate;
89         end if;
90     end if;
91 end process;
92
93 end cntlumbbral_arch;
    
```

Listing 6.3: Scritp encargado del Complemento a dos

```
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3 use IEEE.STD_LOGIC_UNSIGNED.ALL;
4
5 entity complemento_a_dos is
6     Port ( input_bus : in  std_logic_vector(13 downto 0);
7           output_bus : out std_logic_vector(13 downto 0));
8 end complemento_a_dos;
9
10 architecture Behavioral of complemento_a_dos is
11 begin
12
13     process(input_bus)
14     begin
15         if input_bus(13) = '0' then
16             output_bus <= input_bus;
17         else
18             output_bus <= not(input_bus) + '1';
19         end if;
20     end process;
21
22 end Behavioral;
```

Bibliografía

- [1] GAISSER T. *Cosmic rays and particle physics*, 2 ed, Cambridge, Cambridge university press; 2016.
- [2] GRIEDER P. *Cosmic rays at Earth: Researcher´s reference manual and data book*, 1st ed. Amsterdam: Elsevier; 2001.
- [3] LONGAIR M. *High energy astrophysics*, 3 ed, Cambridge, Cambridge university press;2011.
- [4] BERTOLOTTI M. *Celestial messengers: cosmic rays*, springer 2013.
- [5] SCHRÖDER F. *Instruments and methods for the radio detection of high energy cosmic rays*, Springer,Verlag Berlin Heidelberg; 2012.
- [6] PETER K. GRIEDER, *Extensive Air Showers: High Energy Phenomena and Astrophysical Aspects A Tutorial, Reference Manual and Data Book*, Springer Verlag Berlin Heidelberg; 2010.
- [7] KNOLL G. *Radiation Detection and Measurement*, Wiley; 2010.
- [8] STANEV T. *High Energy Cosmic Rays*, 3 ed, Springer Nature Switzerland, springer;2021.
- [9] <https://redpitaya.com/>
- [10] <https://www.ti.com/product/es-mx/LMH5401-SP>
- [11] <https://www.physics.purdue.edu/jones105/phys56400Fall2017/sgc-bc400-404-408-412-416-data-sheet.pdf>
- [12] <https://www.onsemi.com/pdf/datasheet/microj-series-d.pdf>

-
- [13] <https://www.onsemi.com/pub/Collateral/AND9770-D.PDF>
- [14] HAMAMATSU PHOTONICS *Photomultiplier tubes: Basics and Applications*, Third edition, committee; 2006.
- [15] WRIGHT A. *The photomultiplier handbook*, first edition, Oxford University Press y Oxford, United Kingdom; 2017.
- [16] CHARLES D. DERMER and GOVIND MENON *High Energy Radiation from Black Holes: Gamma Rays, Cosmic Rays, and Neutrinos*, Princeton Series in Astrophysics; 2009.
- [17] SCHLICKEISER R., *Cosmic Ray Astrophysics*, Springer Verlag Berlin Heidelberg; 2002.
- [18] ARNALDI, L., CAZAR RAMÍREZ, DENNIS, AUDELO, MARIO, SIDELNIK, IVAN. *The new data acquisition system of the LAGO Collaboration based on the Redpitaya board*. 87-92. 2020.
- [19] AFANASIEV G., *Fundamental Theories of Physics: Vavilov-Cherenkov and Synchrotron Radiation, Foundations and Applications*, Springer; 2010.
- [20] WILLIAM R. LEO, *Techniques for nuclear and particle physics experiments*. Springer, 1992.
- [21] HAKAMATA T. *Photomultiplier tubes Basics and Applications*, Third edition, Word technical Writing, 2006
- [22] J. V. JELLEY, *Čerenkov radiation: Its origin, properties and applications*, Contemporary Physics, 3:1, 45-57; 1961.
- [23] HARRISON N. *Photomultiplier manual: theory, design, application*, RCA corporation; 1970.