



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

DISEÑO E IMPLEMENTACIÓN DE UN DETECTOR DE RADIACIÓN PARA
SATÉLITES ESTÁNDARES CUBESAT 3U

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO

HÉCTOR IGNACIO ZÚÑIGA MIRANDA

PROFESOR GUÍA:
MARCOS DÍAZ QUEZADA

MIEMBROS DE LA COMISIÓN:
CRISTOBAL GARRIDO CÁCERES
ERNESTO GRAMSCH LABRA

SANTIAGO DE CHILE
2022

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO
POR: HÉCTOR IGNACIO ZÚÑIGA MIRANDA
FECHA: 2022
PROF. GUÍA: MARCOS DÍAZ QUEZADA

DISEÑO E IMPLEMENTACIÓN DE UN DETECTOR DE RADIACIÓN PARA SATÉLITES ESTÁNDARES CUBESAT 3U

La exploración y la caracterización del ambiente espacial a un bajo costo es una necesidad latente en el desarrollo tecnológico de estos tiempos. En este contexto, se desarrolla la tecnología CubeSat, un diseño estandarizado que permite el desarrollo de tecnología espacial a un muy bajo costo y que abre una oportunidad a la innovación y a la investigación.

Una de las condiciones más hostiles que se pueden encontrar en el espacio y no en la tierra es la radiación, la cual genera efectos nocivos tanto en las tecnologías como en la vida.

Con esta motivación, en este trabajo se presenta el diseño y la implementación de un detector de partículas de alta energía, con el fin de caracterizar la radiación presente en el espacio a un bajo costo y con un instrumento de tamaño reducido, capaz de acoplarse a un misión de satélites CubeSat.

El dispositivo implementado se compone de una etapa analógica y una digital. La etapa analógica consta de un fotodetector de silicio, el cual interactúa con partículas de alta energía (radiación) y genera una carga proporcional a la energía de la partícula. Luego esta carga se transforma en un pulso de voltaje, el cual es amplificado. El pulso voltaje amplificado se compara con 4 niveles de voltajes diferentes. Estos niveles de voltaje representan un nivel de energía, es decir, si el nivel de voltaje es superado por el pulso, significa que la partícula que generó ese pulso poseía una energía mayor a la energía asociada a ese nivel.

Luego, se realiza un conteo de cuántas partículas superaron cada nivel de voltaje, de esto se encarga la etapa digital. Utilizando 4 contadores de pulsos (uno por cada nivel), un microcontrolador PIC16F877A y una Raspberry Pi Zero Wireless, la etapa digital es capaz de contar cuantas partículas superan cada uno de los niveles de voltajes. La comunicación entre el microcontrolador y la Raspberry Pi se realiza mediante el protocolo I2C.

En síntesis, el instrumento se define como contador de partículas, ya que cuenta la cantidad de partículas que superan 4 niveles de energía en una ventana de tiempo. Los 4 niveles de energía son 56.25 [keV], 225 [keV], 562.5 [keV] y 1.125 [MeV].

Por último, se presentan los resultados obtenidos con el dispositivo. Si bien estos no permiten determinar que el dispositivo mide la radiación, sí permiten comprobar que la electrónica y los softwares implementados están funcionando. Para determinar si el dispositivo cumple su objetivo principal, se propone una serie de pruebas con una fuente radiactiva.

Para Lorena y Héctor

Tabla de Contenido

Índice de Tablas	v
Índice de Ilustraciones	vi
1. Introducción	1
1.1. Identificación y Formulación del Problema	1
1.2. Objetivos del Trabajo de Título	2
1.3. Contenido del documento	3
2. Marco Teórico	4
2.1. Nanosatélites y CubeSats	4
2.2. Principio de medición de la radiación	5
2.2.1. Amplificador de carga	7
2.2.2. Amplificador No Inversor	8
2.3. Protocolo de comunicación I2C	10
2.3.1. Definiciones del protocolo	10
2.3.2. Funcionamiento del algoritmo	12
2.3.3. Clock stretching	13
3. Estado del Arte	14
3.1. BurstCube	14
3.2. A Review of Requirements for Gamma Radiation Detection in Space Using CubeSats	15
4. Diseño del instrumento	16
4.1. Circuito Analógico	17
4.1.1. Simulación del circuito analógico	17
5. Implementación del instrumento	25
5.1. Construcción del hardware	25
5.1.1. Placa PCB	25
5.1.2. Etapas del circuito	26
5.2. Procesamiento digital	30
5.2.1. Contador I2C	30
5.2.2. Driver I2C para Raspberry Pi Zero Wireless	33

6. Pruebas realizadas y resultados	36
6.1. Pruebas sin el sensor	36
6.1.1. Resultados etapa amplificación	37
6.1.2. Resultados etapa comparación	38
6.1.3. Resultados procesamiento digital	41
7. Discusión y Análisis	42
7.1. Simulación vs Implementación Real	42
7.2. Análisis de los resultados	43
7.3. Trabajo Futuro	44
7.3.1. Pruebas con fuente de radiación	44
7.3.2. Detector Gamma	44
8. Conclusión y trabajos futuros	46
8.1. Conclusión	46
Bibliografía	48
ANEXOS	50
Anexo A	50
A1. Esquemáticos del sistema completo	50
Anexo B	52
B1. Código PIC16F877A	52
B2. Código Driver I2C	57
Anexo C	58
C1. Trama I2C Contador	58
Anexo D	59
D1. Tablas de componentes	59

Índice de Tablas

4.1. Nivel de energía asociado a cada nivel de voltaje.	24
5.1. Parámetros relevantes TLC072	27
5.2. Parámetros relevantes MAX6126AASA41	28
5.3. Voltajes de referencia teóricos vs obtenidos	29
5.4. Parámetros relevantes LMV339IPWR	29
5.5. SoC de los distintos modelos de Raspberry	35
6.1. Voltajes de referencia teóricos vs obtenidos vs válidos	38
7.1. Nivel de energía asociado a cada nivel de voltaje.	43
8.1. Tabla de los componentes del circuito	59

Índice de Ilustraciones

2.1. Satélite CubeSat SUCHAI 1. Tomada de SPEL, Universidad de Chile.	5
2.2. Detección de radiación mediante un centellador y un fotomultiplicador. Tomada de Grupen et. al. (201)	6
2.3. Esquemático Amplificador de Carga. Adaptado del diseño del Profesor Ernesto Gramsch	7
2.4. Esquemático Amplificador No Inversor. Adaptado del diseño del Profesor Ernesto Gramsch	8
2.5. Diagrama de conexión entre Fotodetector, Amplificador de Carga y Amplificador No Inversor	9
2.6. Diagrama de conexión I2C. Adaptado de Mankar et. al.	10
2.7. Transferencia de un bit en I2C. Adaptado de I2C-bus specification and user manual	11
2.8. Condiciones de START y STOP. Adaptado de I2C-bus specification and user manual	11
2.9. Transferencia de datos en I2C. Adaptado de I2C-bus specification and user manual	12
2.10. Trama I2C. Adaptado de Mankar et. al.	12
2.11. Diagrama de flujo algoritmo I2C. Adaptado de Mankar et. al.	13
3.1. BurstCube	14
3.2. Detector de rayos gamma	15
3.3. Diagrama de flujo del trabajo	15
4.1. Diagrama de bloques funcional del sistema.	17
4.2. Esquemático contador de partículas.	18
4.3. Esquemático de la simulación del amplificador de carga.	18
4.4. Pulso de entrada.	19
4.5. Simulación Amplificador de Carga.	20
4.6. Esquemático de la simulación del amplificador no inversor.	21
4.7. Simulación Amplificador No Inversor.	21
4.8. Esquemático de la simulación del circuito analógico.	22
4.9. Salida del amplificador no inversor.	22
4.10. Salida comparadores.	23
5.1. Modelo 3D de placa PCB. Tomada de diseño de Cristóbal Garrido.	25
5.2. Placa del dispositivo con las distintas etapas	26
5.3. Etapa de amplificación	27

5.4.	Etapa de voltajes de referencia	28
5.5.	Esquemático divisor de voltajes	28
5.6.	Etapa de comparación	29
5.7.	Etapa de procesamiento digital	30
5.8.	Raspberry Pi Zero Wirless	31
5.9.	Esquema general del Contador I2C. Tomado de Cristián Lopez	31
5.10.	Diagrama de flujo del Contador I2C	32
5.11.	Trama I2C de la función en python	33
6.1.	Diagrama de conexiones para pruebas	36
6.2.	Configuración de pruebas sin sensor	37
6.3.	Salida etapa de amplificación escala 50 [mV]	37
6.4.	Salida etapa de amplificación escala 500 [mV]	38
6.5.	Salida comparador conectado a la entrada B0, nivel 100 [mV]	39
6.6.	Salida comparador conectado a la entrada B1, nivel 400 [mV]	39
6.7.	Salida comparador conectado a la entrada A0, nivel 1 [V]	40
6.8.	Salida comparador conectado a la entrada A1, nivel 2 [V]	40
6.9.	Resultados etapa de procesamiento digital	41
7.1.	Cambio Amplificador No Inversor	42
7.2.	Sensor 7478 en KiCad	45
7.3.	Elementos necesarios para adaptar del detector a un detector de radiación gamma	45
8.1.	Esquemático Circuito Analógico.	50
8.2.	Esquemático Circuito Digital.	51
8.3.	Trama I2C Contador. Tomado de Cristián Lopez	58

Capítulo 1

Introducción

1.1. Identificación y Formulación del Problema

El desarrollo de la tecnología satelital está en crecimiento, gracias al rápido avance de las tecnologías de estado sólido y consigo los circuitos integrados, desde finales de los años 90, comenzaron a re-aparecer los nanosatélites [1] lo cual fue un cambio muy importante en el paradigma de la construcción de los satélites. No sólo se abarataron costos debido a la reducción en tamaño y al uso de componentes electrónicas comerciales para su construcción, sino que se cambió la filosofía de diseño de los satélites, pasando de una filosofía clásica, que consistía en construir el satélite de acuerdo a las necesidades del experimento que este portaría, a una filosofía que propone estandarizar el diseño de los satélites.

Este nuevo paradigma de estandarización dio nacimiento a lo que se conoce como CubeSats, un estándar de diseño de satélites cuya estructura es escalable a cubos de 10 [cm] de arista, lo cual se denomina como una unidad de 1 U, y una masa inferior a los 1,33 [kg][1]. Las ventajas que presentan los nanosatélites y, en particular, los CubeSats son variadas, en primer lugar existe una reducción en los tiempos de desarrollo ya que el diseño es estandarizado, esto tiene un impacto en el costo del satélite, el cual se ve considerablemente reducido por el poco tiempo que necesita para desarrollarlo y por las dimensiones reducidas de los CubeSats. Esta reducción en el costo permite que se puedan tomar riesgos, es decir, se le abre espacio a la innovación y a la investigación, ya que se pueden hacer más experimentos sin perder mucho en caso de que estos no sean exitosos. En cuanto a las desventajas, la principal es que se presenta una limitación espacial importante, por lo que hay que tener esta consideración en los diseños de las cargas y experimentos.

En este contexto, uno de los roles que cumplen los nanosatélites en esta era del Nuevo Espacio (*The New Space Era*), es la de *pathfinders*, es decir, tienen la tarea de encontrar vías o caminos, en los cuales las tecnologías de mayor escala, que se deseen mandar al espacio, sean viables [1]. Esto significa que estos satélites tienen como tarea recolectar información sobre las condiciones que existen en el espacio exterior de tal manera de evaluar cuáles son las consideraciones que se deben tomar en cuenta al mandar una nueva tecnología al espacio. Por lo tanto, una de las necesidades que existe es la de instrumentos capaces de medir las distintas condiciones a las que se enfrentan estos satélites.

Una de estas condiciones a las que se enfrentan las tecnologías que son mandadas al espacio y que no se presenta en la tierra es la exposición a la radiación, esta puede ser muy perjudicial para los experimentos que se llevan en un satélite y para la electrónica de todo el sistema que hace posible el funcionamiento de éste.

La radiación corpuscular, es decir de partículas de alta energía, produce daños en la electrónica de los sistemas, por ejemplo, existe una zona en la magnetósfera llamada Cinturones de Van Allen en la cual residen electrones de alta energía, estos son denominados como *electrones asesinos* y son capaces de generar descargas eléctricas internas, denominadas *Single-Event Effects* [2] lo suficientemente fuertes para dañar la electrónica de los satélites [3].

Por otro lado, la radiación electromagnética produce efectos nocivos en la electrónica, mediante los mismos efectos mencionados anteriormente, y además es tremendamente dañina para la biología, produciendo efectos como cambios en la expresión de los genes, alteraciones en sistemas biológicos del cuerpo humano relacionados con problemas de visión de los astronautas y efectos a largo plazo como el cáncer [4].

Por lo tanto, surge la necesidad de tener un instrumento que permita medir la cantidad de radiación que afecta al satélite y que esta información se pueda utilizar y procesar para hacer un análisis de cómo esta puede llegar a afectar el funcionamiento óptimo, tanto del experimento que se está mandando, como de los sistemas mismos del satélite.

El instrumento diseñado e implementado en la presente memoria corresponde a un contador de partículas, conformado por dos grandes etapas. La primera, una etapa analógica que permite la detección de radiación corpuscular, en particular electrones, y la amplificación de la señal detectada. En segundo lugar, una etapa digital capaz transformar la señal analógica en una digital, realizar un conteo de las partículas que ingresan al sistema y comunicar esta información mediante el protocolo I2C.

1.2. Objetivos del Trabajo de Título

El objetivo general de la memoria de la siguiente manera:

Diseño e implementación de un detector capaz de medir la cantidad de radiación, ya sea corpuscular o electromagnética, que reciben los sistemas y experimentos, tanto naturales como artificiales, que se estén llevando en un satélite de tipo CubeSat.

De este objetivo general se derivan los siguientes objetivos específicos.

En primer lugar, el diseño e implementación del circuito analógico del detector. Es decir, el diseño del esquemático, su simulación en algún software de tipo SPICE y, a través de un software tipo EDA, la implementación de este circuito en una placa PCB la cual se imprimirá y se montará el circuito diseñado en ella.

Un segundo objetivo corresponde a la implementación del procesamiento digital de las señales obtenidas por el circuito analógico, esto implica transformar la señal analógica en una digital y luego, la programación de un microcontrolador para procesar la información y comunicarla mediante I2C.

Como tercer objetivo está la adaptación del detector al satélite. Como se definió anteriormente, el diseño está pensado para un satélite 3U, por la cual se debe adaptar a esas dimensiones. Esto implica tomar en consideración las limitaciones de espacio, la cara en donde se desea que el detector se posicione y también se debe tomar en cuenta las necesidades de los experimentos, tanto en cuanto a espacio como posición. Además, implementar un software capaz de comunicar con el instrumento con el software del satélite.

Por último, se define como cuarto objetivo la caracterización del instrumento, es decir, establecer los rangos de medición de este, su resolución y las limitaciones que este puede tener. Este punto implica el desafío de conseguir una fuente estable de rayos gamma en donde se puedan realizar las pruebas al detector.

1.3. Contenido del documento

En el Capítulo 2, se presentará un marco teórico en donde se contextualiza el problema abordado, se justifica teóricamente el uso de los circuitos implementados en el instrumento y se presenta el protocolo de comunicación utilizado para comunicar el instrumento con otros sistemas.

En el Capítulo 3, se presenta una investigación bibliográfica en donde se revisan otros sistemas de medición de radiación que se han implementado en satélites CubeSat.

Posteriormente, en el Capítulo 4, se presenta el diseño del instrumento, haciendo énfasis en las simulaciones del circuito analógico, la explicación del funcionamiento de las distintas etapas de este y cómo estas interactúan entre sí.

En el Capítulo 5, se procede a describir la implementación del instrumento, pasando por la construcción del hardware, mostrando sus distintas etapas y componentes, y por otro lado la implementación del procesamiento digital, el cual consta de dos partes, la primera parte es el Contador I2C, el cual forma parte del instrumento y permite el conteo de las partículas y la comunicación de esta información mediante el protocolo I2C. La segunda parte consta del software que permite la integración del instrumento con el satélite. Este se programa en una Raspberry Pi y es el que funciona como un *driver* del instrumento para permitir obtener la información del detector al sistema central del satélite.

En el Capítulo 6, se proceden a mostrar las pruebas realizadas al instrumento y los resultados obtenidos de estas. Luego, en el Capítulo 7 se discuten estos resultados y se realiza una discusión sobre las diferencias entre el diseño teórico y el implementado, los resultados obtenidos y el trabajo a futuro que se debe realizar con este proyecto.

Por último en el Capítulo 8 se realiza una conclusión analizando el cumplimiento de los objetivos y rescatando los resultados obtenidos.

Capítulo 2

Marco Teórico

2.1. Nanosatélites y CubeSats

Los satélites CubeSats corresponden a un estándar de diseño de nanosatélites cuya estructura es escalable a cubos de 10 [cm] de arista, lo cual se denomina como una unidad de 1U, y una masa inferior a los 1,33 [kg] [1]. En particular, el diseño del presente trabajo pretende ser adaptable a un CubeSat de 3U, es decir, tres cubos de 10 cm de arista, los cuales conforman un paralelepípedo de $30 \times 10 \times 10$ [cm].

En cuanto a su historia y el porqué de su importancia, la idea del diseño del CubeSat fue propuesta en 1999 por el profesor el profesor Robert Twiggs de Stanford, sin embargo, no fue hasta el año 2003 cuando se realizaron los primeros lanzamientos de estos satélites. En este año se marcó una revolución sin precedentes en la evolución de los pequeños satélites, pasando a conformarse la Era CubeSat [1].

Pumpkin Inc. en San Francisco, comenzó a comercializar kits CubeSats, lo cual permitió un amplio esparcimiento de este diseño en distintas instituciones y, por lo tanto, un acelerado proceso de desarrollo en innovación en torno a los CubeSats en distintas partes del mundo. En particular, en el año 2017 se lanzó un total de 312 CubeSats a la órbita espacial. Lo anterior se explica por un crecimiento en la capacidad de lanzamiento de los satélites, debido a fondos gubernamentales destinados a laboratorios y universidades dedicadas al desarrollo de estos satélites y a empresas privadas tratando de proveer nuevas tecnologías de comunicación y nuevas capacidades de observación de la tierra utilizando CubeSats [1].

En cuanto a la instrumentación y los CubeSats, existe un escenario de gran desarrollo e innovación en cuanto instrumentos para medir cosas tanto dentro como fuera de la tierra. Por ejemplo, para el satélite HaloSat se construyó un sensor de rayos X para medir fuentes de emisión difusas tales como la radiación del halo gaseoso de la vía lactea [5]. Por otro lado está el BurstCube, un CubeSat destinado a medir *Gravitational Wave Counterparts* que corresponden a brotes de rayos gamma (GBR) que se forman como contraparte de las ondas gravitacionales [6]. En cuanto a mediciones terrestres se tiene como ejemplo un CubeSat de la Universidad de Washington que tiene como objetivo realizar mediciones para tener un modelo de la ionósfera [7] y, por otro lado, un instrumento a escala de CubeSats para medir

la cantidad de hielo en las nubes[8].

En relación a lo anterior, en el presente trabajo es de particular interés desarrollar un instrumento adaptable a un satélite CubeSat. Esto presenta ciertas limitaciones, principalmente económicas y de espacio. Por lo tanto, se debe desarrollar una electrónica pequeña con componentes de bajo costo y las dimensiones el instrumento deben ser de tal manera de que se pueda adaptar a un satélite de 3U.

Con lo anterior en mente, se diferencia el presente trabajo de los ejemplos mencionados anteriormente en que el detector de radiación está pensado para adaptarse a un satélite cualquiera y no ser ni la carga, ni el experimento principal del satélite. El objetivo es que se utilice como indicador de las condiciones a las que se enfrentan los experimentos que lleva el satélite al que acompaña.

Para lograr este objetivo, sin embargo, es necesario entender los principios de detección de la radiación, entender la electrónica necesaria para poder trabajar con las señales medidas y, por último, estudiar una manera de comunicar la información. Estos temas se desarrollarán en las secciones siguientes.

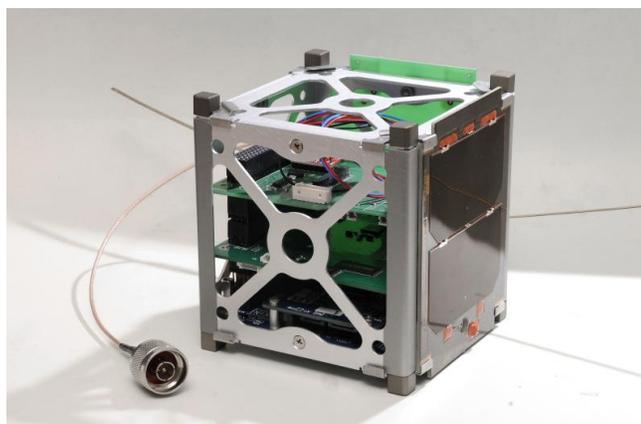


Figura 2.1: Satélite CubeSat SUCHAI 1. Tomada de SPEL, Universidad de Chile.

2.2. Principio de medición de la radiación

Al momento de diseñar un instrumento es esencial entender qué es lo que se desea medir. En este caso el objetivo es medir la cantidad de radiación que afecta a un satélite en el espacio. Existen dos tipos de radiación, la electromagnética y la corpuscular, en particular, es de particular interés medir la radiación gamma y la radiación de electrones.

La radiación de electrones corresponde a electrones de alta energía atrapados en anillos de radiación alrededor de la Tierra. Estas partículas alcanzan velocidades cercanas a la de la luz, por lo cual transportan una gran cantidad de energía, de hecho, un electrón puede tener cerca de 1000 veces más energía que la radiación de rayos X de una radiografía [3].

Estos electrones residen en lo que se conoce como el Cinturón de Radiación de Van Allen [3], la cual es una zona de la magnetosfera. Esta se encuentra a una altitud entre los 13000 y 19000 [km], en donde se encuentra una gran cantidad de satélites tales como los de GPS.

Sin embargo, debido a la intensa actividad solar, estos pueden ser empujados a zonas más cercanas a la tierra, dañando a aún más satélites.

El principio de detección es el siguiente: en el caso de partículas de alta energía, en este caso electrones, estos interactúan con los electrones del material semiconductor que conforma el detector. Al interactuar, los electrones de alta energía depositan su energía formando pares electrón-hueco (e-h). Esta interacción ocurre hasta que la energía de la partícula es muy baja y no sigue produciendo pares e-h [9].

Para los distintos materiales, la energía promedio necesaria para generar un par para generar un par e-h hueco es diferente, sin embargo para un detector de silicio esta energía es igual a 3.6 [eV] [9]. Al generar un par e-h, se induce una carga en el detector igual a la carga de un electrón (e), es decir $1,6 \times 10^{-19}$ [C].

En cuanto a la radiación gamma es un tipo de radiación electromagnética que se genera a partir de procesos radioactivos, procesos subatómicos y también por fenómenos astrofísicos de gran magnitud [10]. Las ondas de rayos gamma se emiten en frecuencias mayores a los 3 [EHz], es decir, 3×10^{18} [Hz], esto, a su vez, corresponde a longitudes de onda del orden de los picómetros, es decir, 10^{-12} [m] [11].

La medición de estos rayos se realiza mediante centelladores, estos son materiales los cuales reaccionan a la radiación a través de fotones. Cuando la radiación interactúa con material, este libera fotones en longitudes de onda del espectro visible y ultravioleta. Estas ondas de menor frecuencia son posibles de medir a través de fotodiodos y fotomultiplicadores, dispositivos electrónicos que al captar una onda de luz generan una corriente eléctrica [12].

En síntesis, partículas de alta energía, ya sea electrones o fotones, interactúan con los detectores generando una señal eléctrica proporcional a la energía de la partícula.

Para amplificar la señal eléctrica generada por un fotodiodo se pueden utilizar dos configuraciones de amplificadores. La primera de ellas es un amplificador de transimpedancia, el cual en su salida entrega un voltaje directamente proporcional a la corriente generada por el fotodiodo. La segunda configuración es la de un amplificador de carga, el cual en su salida entrega un voltaje directamente proporcional a la carga generada por el fotodiodo. Esta segunda configuración es la que se utiliza ya que interesa conocer la cantidad de electrones que entraron al circuito. A partir de esta información se puede conocer la energía de la partícula que interactuó con el fotodiodo y, por ende, la cantidad de radiación.

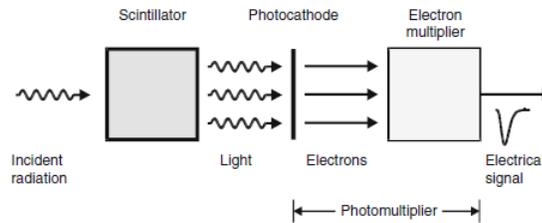


Figura 2.2: Detección de radiación mediante un centellador y un fotomultiplicador. Tomada de Grupen et. al. (201)

2.2.1. Amplificador de carga

A continuación, se presenta el estudio analítico de un amplificador de carga y la relación matemática entre el voltaje de salida y la carga de entrada. En la figura 2.3 se presenta el esquemático de un amplificador de carga, cabe destacar que se reemplaza el fotodiodo por una fuente de corriente.

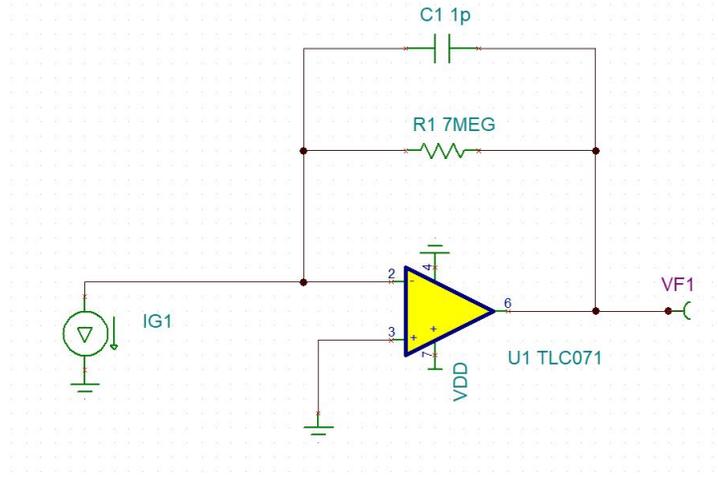


Figura 2.3: Esquemático Amplificador de Carga. Adaptado del diseño del Profesor Ernesto Gramsch

Aplicando las fórmulas conocidas para un amplificador operacional, se obtienen las ecuaciones presentadas en 2.1. $IG1$ corresponde a un generador de corriente que reemplaza al fotodetector para efectos de la simulación que se presentará más adelante. $VF1$ corresponde al voltaje de salida, es decir, un voltaje proporcional a la carga de la entrada. $R1$ y $C1$ son una resistencia y un condensador, respectivamente, que determinan la ganancia del circuito.

$$\begin{aligned}
 \frac{VF1}{R1//C1} &= IG1 \\
 VF1 &= IG1 \cdot (R1//C1) \\
 VF1 &= IG1 \cdot \frac{\frac{R1}{C1s}}{R1 + \frac{1}{C1s}} \\
 VF1 &= IG1 \cdot \frac{R1}{R1 \cdot C1s + 1} \\
 VF1 &= IG1 \cdot \frac{1}{C1s + \frac{1}{R1}} \tag{2.1} \\
 \frac{1}{R1} &\sim 0 \\
 \Rightarrow VF1 &= IG1 \cdot \frac{1}{C1s} \\
 VF1 &= \frac{IG1}{C1s} \\
 VF1 &= \frac{Q}{C1}
 \end{aligned}$$

De las ecuaciones se concluye que el voltaje de salida $VF1$ es directamente proporcional a la carga de entrada Q del circuito. Por otro lado, la ganancia de este amplificador está determinada por el condensador $C1$ en el loop de retroalimentación, en particular, es inversamente proporcional al valor de este condensador.

2.2.2. Amplificador No Inversor

Dado que la señal generada por amplificador de carga es muy pequeña, es necesario incluir en el diseño una etapa de amplificación de esta señal. Para ello, se agrega un bloque en el circuito analógico compuesto por un Amplificador No Inversor. A continuación, se realiza un estudio analítico de esta configuración en las ecuaciones 2.2 y se presenta el esquemático en la figura 2.4

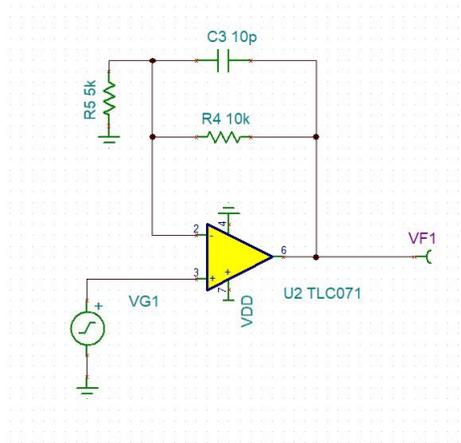


Figura 2.4: Esquemático Amplificador No Inversor. Adaptado del diseño del Profesor Ernesto Gramsch

La fuente de voltaje $VG1$ corresponde a la entrada del amplificador y representa la señal que viene de la salida del amplificador de carga, es decir, $VF1$ de la figura 2.3. $R4$ y $R5$ son resistencias que determinan la ganancia del amplificador. $C3$ corresponde a un condensador que determina las frecuencias que se filtrarán de la señal de entrada en conjunto con $R4$. Aplicando las fórmulas conocidas para un amplificador operacional se obtienen las ecuaciones presentadas en 2.2.

$$\begin{aligned}
\frac{VF1 - VG1}{R4//C3} &= \frac{VG1}{R5} \\
\frac{VF1}{R4//C3} &= VG1 \cdot \left(\frac{1}{R4//C3} + \frac{1}{R5} \right) \\
VF1 &= VG1 \cdot (R4//C3) \cdot \left(\frac{R5 + R4//C3}{R4//C3 \cdot R5} \right) \\
VF1 &= VG1 \cdot \left(1 + \frac{R4//C3}{R5} \right) \\
VF1 &= VG1 \cdot \left(1 + \frac{\frac{R4}{sC3}}{\left(R4 + \frac{1}{sC3} \right) \cdot R5} \right) \\
VF1 &= VG1 \cdot \left(1 + \frac{R4}{s \cdot R4 \cdot C3 + R5} \right)
\end{aligned} \tag{2.2}$$

Cabe destacar que, desde el punto de vista teórico, la configuración corresponde a un filtro activo pasa-baja, sin embargo, al estar trabajando con señales de baja frecuencia la configuración funciona como un amplificador no inversor. En este sentido, al hacer un análisis DC de las ecuaciones y el circuito, se obtiene que la ganancia del circuito está dada por G presentada en la ecuación 2.3. En cuanto a la frecuencia de corte, esta se obtiene con la última línea de la ecuación 2.2 volviendo al dominio del tiempo, con esto, se presenta la frecuencia de corte f_c presentada en la ecuación 2.3.

$$\begin{aligned}
G &= \frac{R4}{R5} \\
f_c &= \frac{1}{2\pi R4 \cdot C3}
\end{aligned} \tag{2.3}$$

Por último, la manera en que se conectan este circuito con el de la sección anterior se muestra en la figura 2.5. En ella se observa que a la entrada del amplificador de carga se conecta el detector que interactúa con la radiación. Luego, el detector convierte esta interacción en una señal eléctrica, la cual es transformada a una señal voltaje por el amplificador de carga. Luego, la señal de voltaje es amplificada por el amplificador no inversor, con el fin de obtener una señal con la que sea posible trabajar en la sección digital.

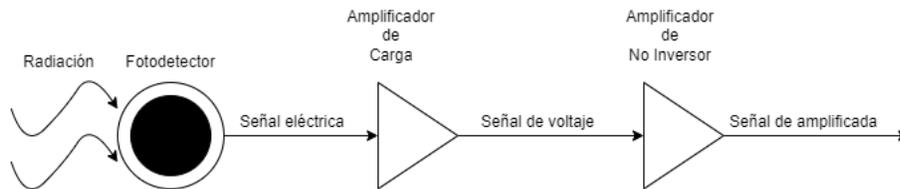


Figura 2.5: Diagrama de conexión entre Fotodetector, Amplificador de Carga y Amplificador No Inversor

2.3. Protocolo de comunicación I2C

El protocolo *Inter Integrated Circuit* o más conocido como I2C, es un protocolo de comunicación serial que provee una interfaz sencilla de usar, sin pérdida de datos y alta velocidad en comparación a otros protocolos de comunicación serial. Este estándar de comunicación fue desarrollado por la empresa Phillips a principios de la década de los 80 con el fin de facilitar la comunicación entre varias componentes en un mismo circuito [13].

El protocolo requiere de dos líneas de datos además de una conexión a tierra para realizar la comunicación. La primera de ellas se denomina *Serial Data* (SDA) y cumple la función de transportar la información. La segunda línea se denomina *Serial Clock* (SCL) y funciona como un reloj de sincronización. Estas líneas tienen la característica de ser *halfduplex*, es decir, pueden transmitir en ambas direcciones pero sólo en una dirección a la vez [14].

Una de las ventajas del protocolo I2C es la posibilidad de conectar varios dispositivos en paralelo a la misma línea de comunicación, cada dispositivo se reconoce a través de una dirección única y estos pueden operar tanto de transmisor como de receptor [13].

2.3.1. Definiciones del protocolo

En un sistema I2C existen dispositivos maestros y esclavos. En su forma más simple, existe sólo un dispositivo maestro y varios dispositivos esclavos los que se conectan como se muestra en el diagrama de la figura 2.6.

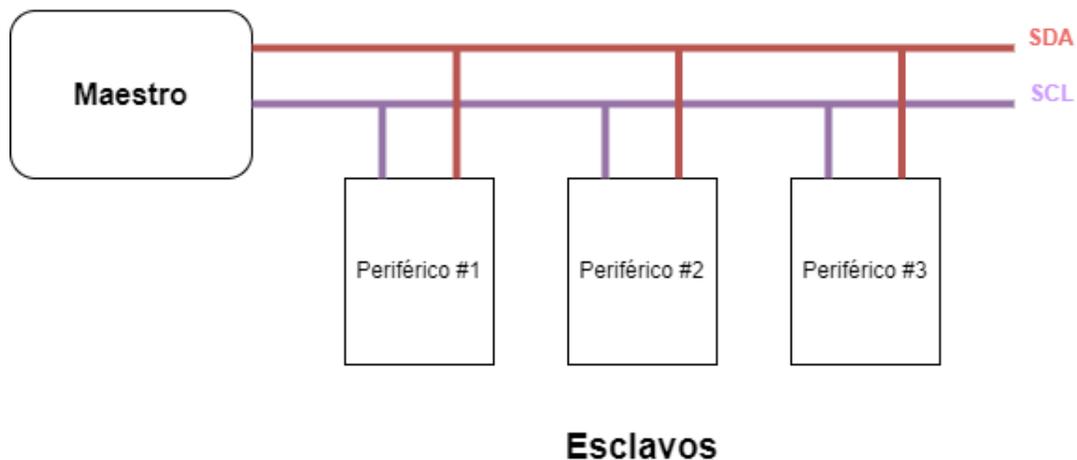


Figura 2.6: Diagrama de conexión I2C. Adaptado de Mankar et. al.

Como se mencionó anteriormente, las líneas SDA y SCL son bidireccionales y están conectadas a una fuente de alimentación positiva a través de una resistencia de *pull-up*. Cuando el bus está libre, ambas líneas están en un nivel lógico alto o HIGH [14].

Los datos en un bus I2C pueden ser transferidos a velocidades hasta 100 [kbits/s] en el modo estándar, hasta 400 [kbits/s] en el modo rápido (*Fast mode*), 1 [Mbit/s] en el modo *Fast-mode Plus* y hasta 3.4 [Mbit/s] en el modo *High-speed*. La capacitancia del bus limita el número de esclavos que se pueden conectar al bus y también define su velocidad [15].

Debido a la gran variedad de tecnologías semiconductoras, el voltaje de los niveles lógicos no está definido sino que depende del nivel de V_{DD} asociado a la tecnología.

En cuanto a los datos transmitidos por la línea SDA, estos deben estar estables durante el periodo HIGH del reloj transmitido en la línea SCL, de esta manera, el estado HIGH o LOW de la línea de datos sólo puede cambiar cuando la línea SCL está en LOW.

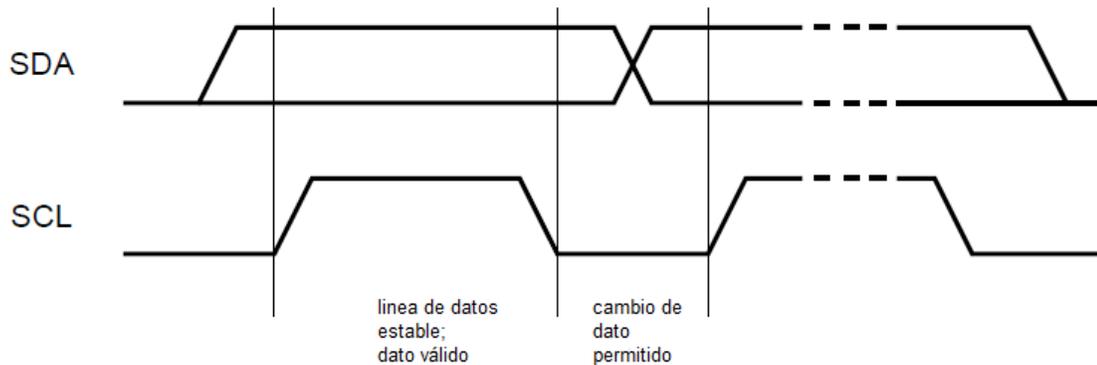


Figura 2.7: Transferencia de un bit en I2C. Adaptado de I2C-bus specification and user manual

Todas las transacciones de datos comienzan con un bit de inicio llamado START (S) y terminan con un bit STOP (P). Una transición de HIGH a LOW en la línea SDA mientras la línea SCL está en HIGH, define la condición de inicio START. Por otro lado Una transición de LOW a HIGH en la línea SDA mientras la línea SCL está en HIGH, define la condición de detención STOP.

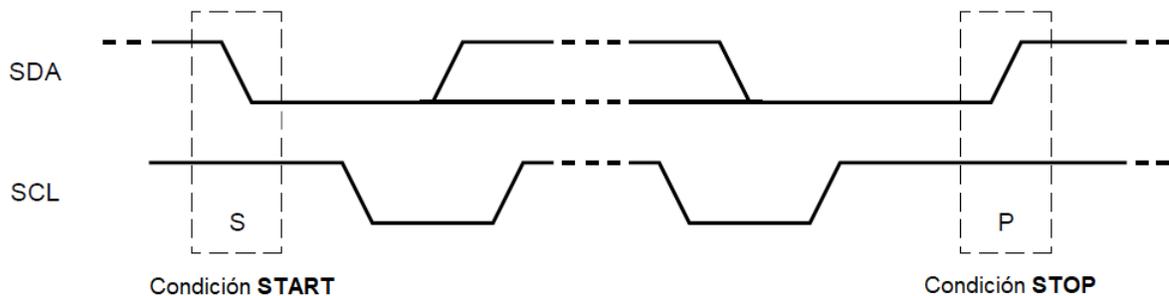


Figura 2.8: Condiciones de START y STOP. Adaptado de I2C-bus specification and user manual

Los bits START y STOP son siempre generados por el dispositivo maestro. Se considera que el bus está ocupado después de la condición START, por otro lado, se considera que el bus está libre luego de un cierto tiempo después de la condición STOP.

Cada byte transmitido en la línea SDA debe ser de 8 bits. El número de bytes que pueden

ser transmitidos es indefinido. Cada byte debe ir seguido por un bit de *Acknowledge* (ACK). Los datos se transfieren en el formato del bit más significativo (MSB).

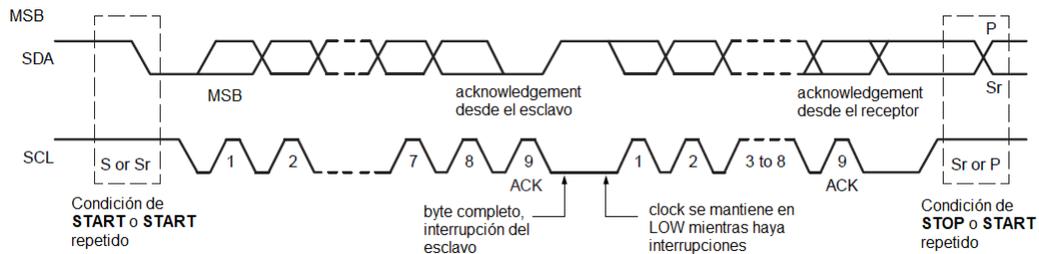


Figura 2.9: Transferencia de datos en I2C. Adaptado de I2C-bus specification and user manual

El bit de *acknowledge* se ubica al final de cada byte transmitido, este permite señalar al dispositivo que está transmitiendo que el byte fue recibido correctamente por el receptor, y por lo tanto se puede mandar el siguiente byte.

La señal de ACK se define de la siguiente manera: el transmisor libera la línea SDA durante el pulso de reloj correspondiente al bit ACK, de tal manera de que el receptor pueda llevar la línea SDA al estado LOW y permanecer estable en LOW durante el periodo HIGH del reloj.

Se define como señal *Not Acknowledge* cuando la línea SDA se mantiene HIGH durante nueve pulsos de reloj. Cuando se genera esta señal, el dispositivo maestro puede generar una señal de STOP para abortar la transferencia o repetir una señal de START para comenzar una nueva transferencia.

2.3.2. Funcionamiento del algoritmo

El algoritmo del protocolo se define en el diagrama presentado en la figura 2.11, por otro lado en la figura 2.10 se observa cómo se visualiza la trama I2C.

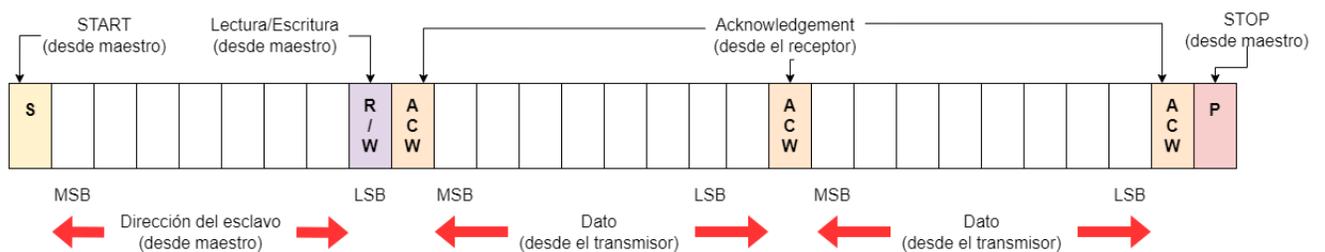


Figura 2.10: Trama I2C. Adaptado de Mankar et. al.

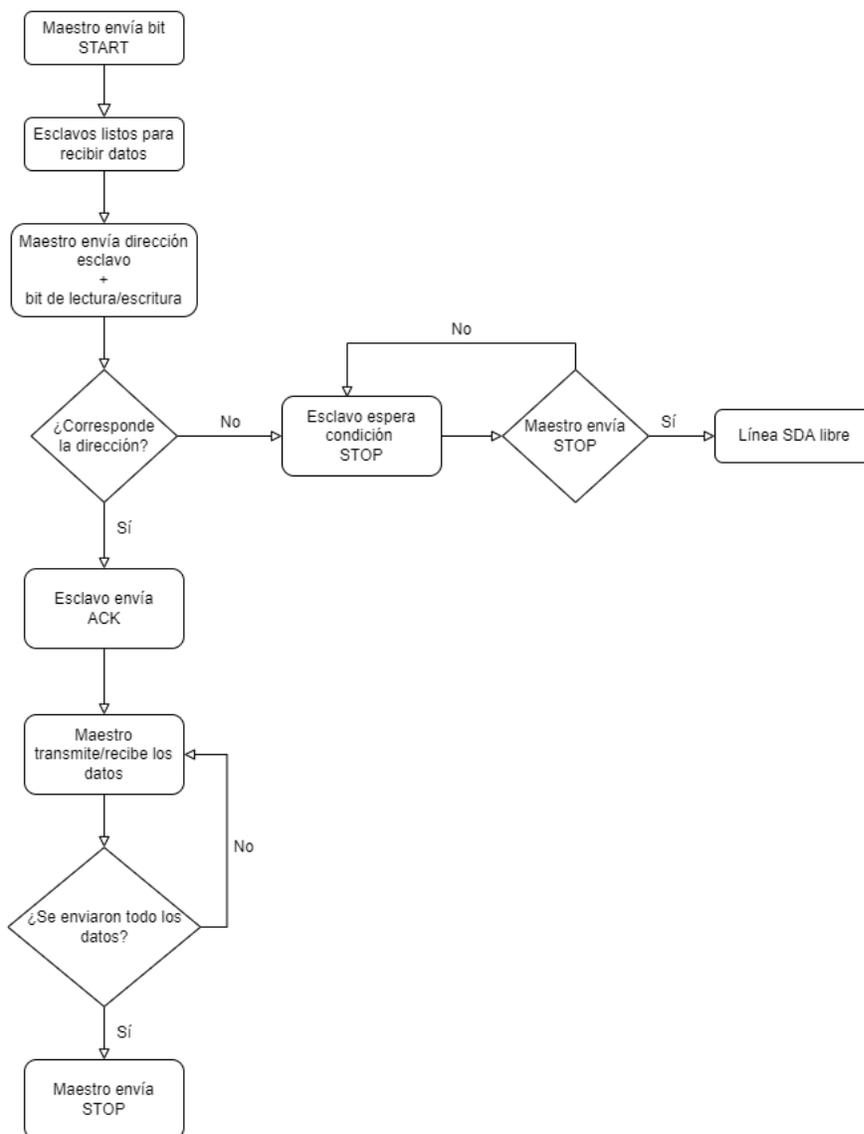


Figura 2.11: Diagrama de flujo algoritmo I2C. Adaptado de Mankar et. al.

2.3.3. Clock stretching

Se define como *Clock stretching* cuando se pausa la transacción de datos al mantener la línea SCL en el estado LOW. Esta transacción no puede continuar hasta que la señal SCL sea liberada y pase a HIGH. Este procedimiento es opcional, de hecho, la mayoría de los dispositivos esclavos no incluyen un driver que permita realizar *clock stretching* [15].

Los beneficios de realizar este mecanismo son asegurar la transferencia de datos y transmitir datos a una mayor velocidad, sin embargo, se necesita más tiempo para almacenar el byte recibido y preparar otro byte para ser transmitido.

Capítulo 3

Estado del Arte

3.1. BurstCube

El BurstCube [6] es un satélite de tipo CubeSat que detectará y localizará brotes de rayos gamma (GRB), que se le atribuyen al choque entre estrellas. Se predice que estos brotes son una contraparte de las ondas gravitacionales. El BurstCube contiene 4 centelladores CsI con un arreglo compacto de fotomultiplicadores de silicio de baja potencia en un CubeSat de 6U. La configuración final sería una constelación de 10 BurstCubes que entreguen una cobertura de todo el cielo.

A diferencia de lo propuesto en la memoria, este es un satélite destinado, únicamente, a medir la radiación gamma, por lo que su “payload” o su carga en totalidad esta destinada a este propósito. En este sentido, se puede tomar como referencia para analizar el principio de funcionamiento de su detector para medir la radiación gamma pero el instrumento que se desea realizar se debe ser capaz de acoplarse a cualquier cubesat de 3 U, lo cual es un tamaño menor al burstcube, y debe ser un periférico más que la carga principal del satélite.

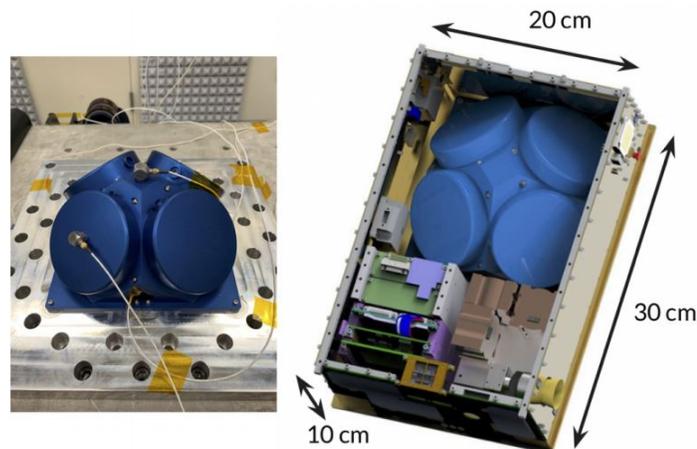


Figura 3.1: BurstCube

3.2. A Review of Requirements for Gamma Radiation Detection in Space Using CubeSats

El trabajo presentado en este paper [16], presenta una revisión sobre la detección de rayos gamma en el espacio por medio de instrumentos a bordo de un satélite de tipo CubeSat 2U o superiores. El estudio presenta las especificaciones y las restricciones que debe tener un satélite que desee llevar un detector de estas características, describiendo, por ejemplo, qué consideraciones se deben tomar para implementar el sistema mecánico, el sistema de potencia, los sistemas de comunicación, además de las características que debe tener el computador a bordo, entre otros.

En cuanto al detector de rayos gamma, este se presenta en la figura 3.2 y corresponde a un diseño que calza en un volumen 1U. Por otro lado, en cuanto la detección de la señal, se presenta el diagrama de flujo de la figura 3.3, el cual consta de etapas de detección, amplificación, digitalización de la señal y procesamiento de los datos.

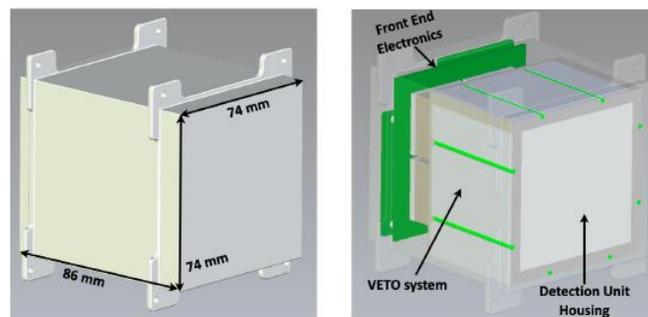


Figura 3.2: Detector de rayos gamma

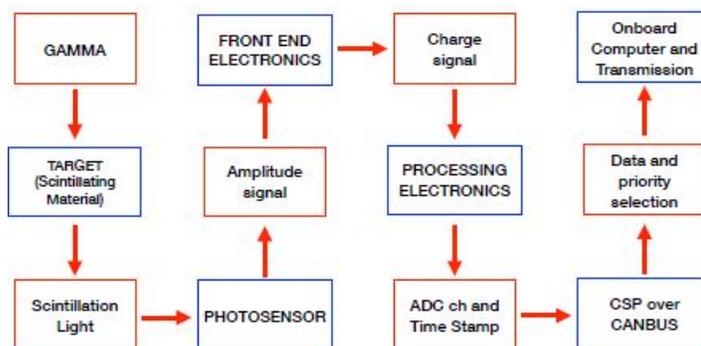


Figura 3.3: Diagrama de flujo del trabajo

Este trabajo sirve como un muy buen marco de referencia sobre las consideraciones que se deben tener en cuanto a cómo acondicionar el satélite para llevar el instrumento y además presenta un sistema de detección muy similar al propuesto en la memoria, como se verá en secciones posteriores.

Capítulo 4

Diseño del instrumento

En primer lugar, se presenta un diagrama de bloques funcional del sistema completo en la figura 4.1. Este funciona de la siguiente manera: en primer lugar, las partículas interactúan con el fotodiodo, el cual genera una pequeña señal eléctrica, tal como se explicó en el capítulo del marco teórico y se representa en la figura 2.2.

La señal eléctrica generada por el fotodetector es amplificada por la primera etapa del circuito analógico, correspondiente a un amplificador de carga, el cual transforma la carga en un voltaje. Posteriormente la señal de voltaje es amplificada por la segunda etapa del circuito correspondiente a un amplificador no inversor.

Una vez amplificada la señal, ésta pasa a un arreglo de 4 comparadores en paralelo. Los comparadores tienen en su salida una salida alta de 5 [V] o baja de 0 [V] de acuerdo si la señal que recibieron supera cierto nivel de voltaje, generando así pulsos cuadrados. Estos pulsos a la salida de los comparadores corresponden al final de la etapa analógica.

En cuanto a la etapa digital, ésta recibe comienza con dos contadores dobles, los cuales reciben los pulsos cuadrados de los comparadores. El contador A y el contador B, cuentan en su interior con dos contadores cada uno. Esto implica que se tienen cuatro contadores de pulsos cuadrados, los cuales se denominarán $A0$, $A1$, $B0$ y $B1$. Cada uno de estos, está asociado a un comparador, es decir, cuenta cuántos pulsos cuadrados se producen a la salida de cada comparador. Esto implica, que los contadores cuentan la cantidad de señales que superaron el nivel de voltaje asociado a su comparador. Este conteo es guardado en los registros de los contadores en 16 bits. Estos son registros son enviados a un microprocesador PIC16F877A. En otras palabras, se cuentan la cantidad de partículas que superan cada nivel de voltaje.

El microprocesador cuenta con un software que solicita la cuenta que llevan los contadores hasta el momento, recibe estas cuentas y las envía mediante el protocolo de comunicación I2C. Como se mencionó en la sección 2.3, para implementar el protocolo de comunicación I2C, se necesita de un dispositivo maestro y uno esclavo, los cuales están comunicados mediante las líneas SDA y SCL. En este sentido, el microprocesador funciona como el dispositivo esclavo y una Raspberry Pi Zero W funciona como el dispositivo maestro.

Finalmente, la Raspberry Pi Zero W cuenta con un software que permite la comunicación con el microprocesador mediante I2C. Éste solicita la información de las cuentas y las recibe para luego almacenarlas, procesarlas y presentarlas al usuario.

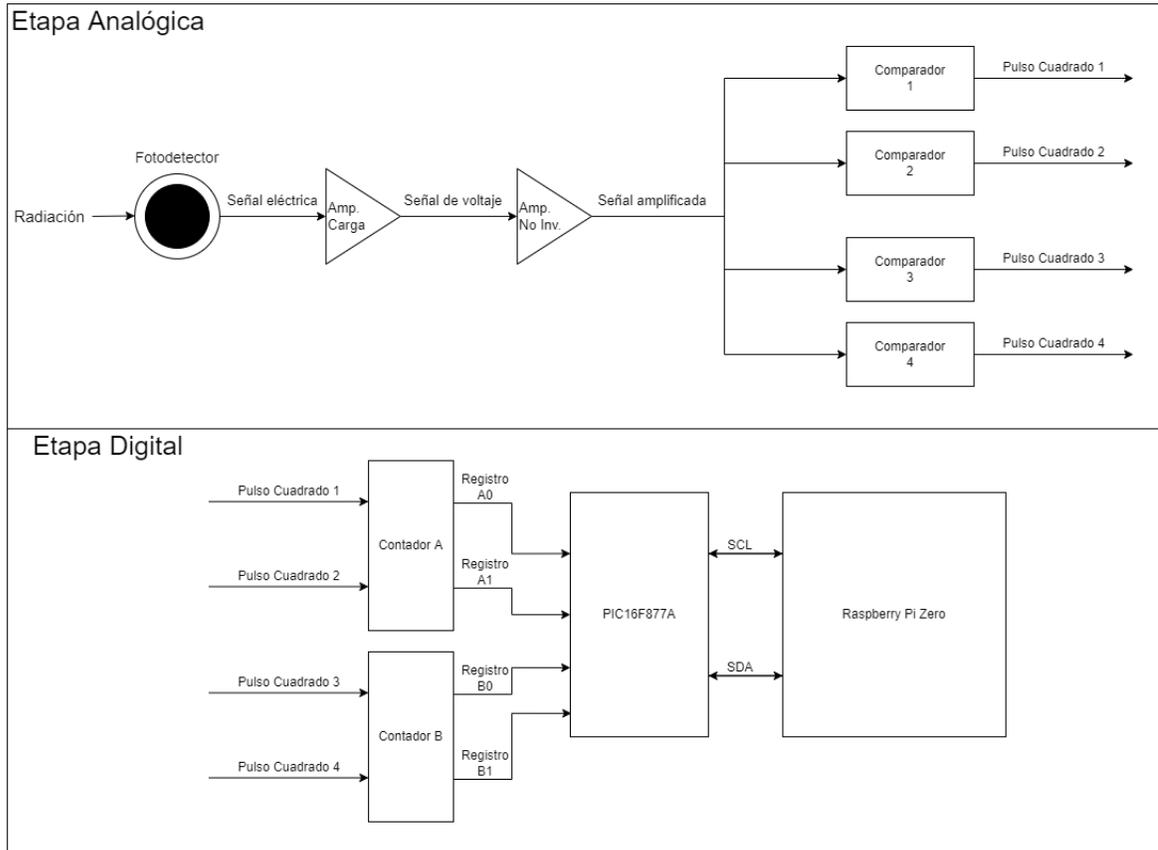


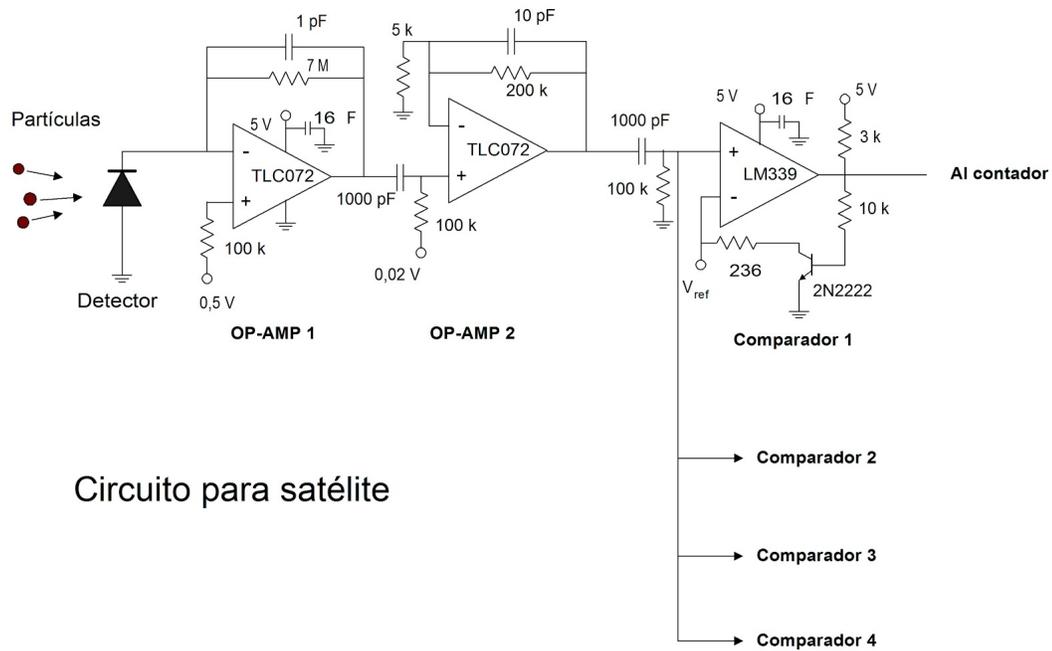
Figura 4.1: Diagrama de bloques funcional del sistema.

4.1. Circuito Analógico

El diseño preliminar del circuito analógico fue presentado por el profesor Ernesto Gramsch el cual se presenta en la figura 4.2. Se tomó como referencia este circuito para realizar el diseño de la sección analógica del detector.

4.1.1. Simulación del circuito analógico

A continuación se presentan las simulaciones realizadas para cada bloque del circuito analógico. Éstas fueron realizadas en el software de diseño SPICE Tina-TI. Se ocupa este software debido a que es necesario la utilización del OpAmp TLC07X, dado que las señales de entrada son pulsos de corriente muy cortos, del orden de los 100 [ps]. Se necesita un opamp lo suficientemente rápido para captar la señal, es decir, que su parámetro *Slewrates* sea alto. Este es el caso del OpAmp TLC07X, cuyo modelo SPICE sólo se encuentra en el software mencionado anteriormente.



Circuito para satélite

Figura 4.2: Esquemático contador de partículas.

Simulación Amplificador de Carga

Para la simulación del amplificador de carga se utilizó el esquemático presentado en la figura 4.3. Es importante destacar el voltaje DC en la entrada positiva del amplificador operacional, esto se debe a que el amplificador no es sensible a señales de niveles tan bajos, por lo tanto es necesario aplicar un voltaje DC para subir el nivel del offset de la señal de entrada.

Por otro lado, de acuerdo a los análisis realizados en el marco teórico en la ecuación 2.1, se puede concluir que la ganancia del amplificador de carga está dada por la ecuación 4.1.

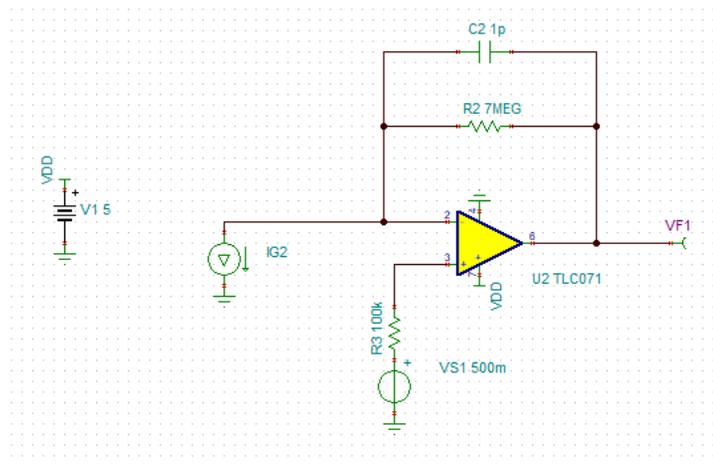


Figura 4.3: Esquemático de la simulación del amplificador de carga.

$$G_{AC} = 1 \times 10^{12} \frac{V}{Q} \quad (4.1)$$

Para la señal de entrada, se simula un pulso de entrada de amplitud $A = 88 \text{ } [\mu\text{A}]$ y periodo $T = 500 \text{ ps}$. El razonamiento para determinar esta corriente de entrada es el siguiente: una partícula necesita una energía de $3,6 \text{ eV}$ para generar un electrón en el detector, también se conoce que los rangos de energía en la entrada del sensor es entre los 100 keV y 1 MeV . Con esta información para ajustar la ganancia de los amplificadores se utiliza el límite superior (1 MeV), para no saturar los amplificadores. Con esto se tiene que con una energía de 1 MeV (al dividirla por $3,6 \text{ eV}$) se generan 277.777 electrones, lo que corresponde a una carga de $-4,4 \times 10^{-14}$ correspondiente al producto de la cantidad de electrones por la carga de un electrón $-1,6 \times 10^{-19}$, por lo tanto, fijando el ancho del pulso en 500 ps , y considerando que $I = \frac{Q}{t}$, se tiene que la amplitud del pulso se calcula como:

$$\begin{aligned} I &= \frac{Q}{t} \\ \Rightarrow I &= \frac{4,4 \times 10^{-14} [C]}{500 [ps]} \\ I &= 88 [\mu A] \end{aligned} \quad (4.2)$$

Con esto, se obtiene la señal de entrada se presenta en la figura 4.4. En ella se observa un pulso cuadrado de amplitud igual a los $88 \text{ } [\mu\text{A}]$ calculados en la ecuación 4.2 y un ancho de 500 ps .

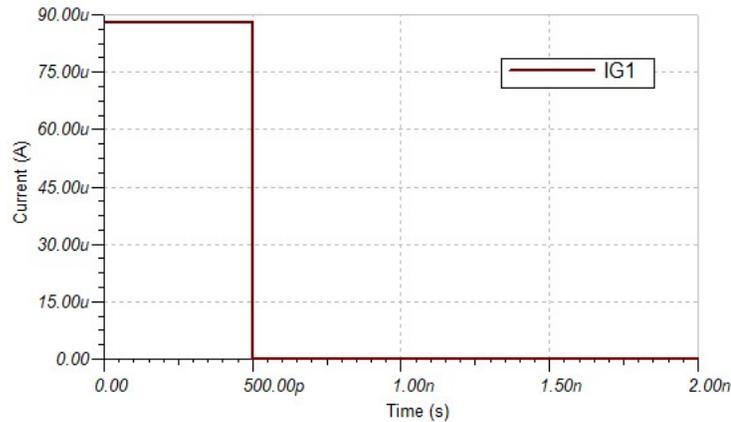


Figura 4.4: Pulso de entrada.

Con esta señal de entrada, se obtienen los resultados presentados en la figura 4.5 . Como se puede apreciar, la señal tiene un offset de salida de 500 mV , el cual fue explicado anteriormente, por otro lado, la amplitud del pulso de salida es de 44 mV , lo cual era esperado considerando la ganancia y la carga de entrada. Éste último análisis se ve expresado en las ecuaciones 4.3.

$$V = 4,4 \times 10^{-14} [C] \cdot 1 \times 10^{12} \left[\frac{V}{C} \right] \quad (4.3)$$

$$V = 44 [mV]$$

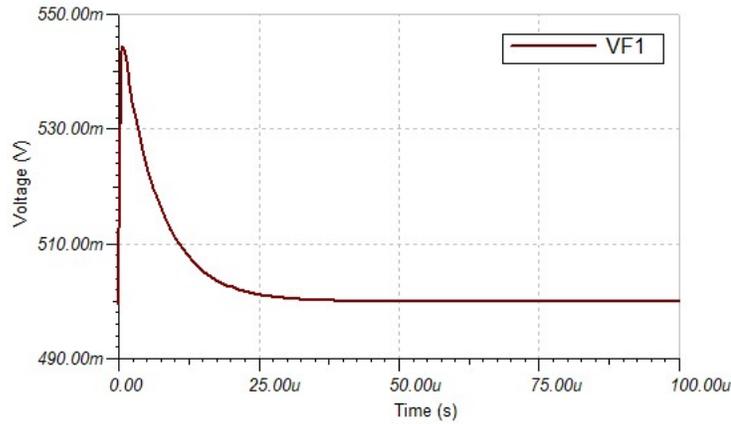


Figura 4.5: Simulación Amplificador de Carga.

Simulación Amplificador No Inversor

Para la simulación del bloque amplificador no inversor, se utilizó el esquemático presentado en la figura 4.6. Estos corresponden a $R4 = 100[k\Omega]$, $R5 = 5[k\Omega]$ y $C3 = 10[pF]$. Con estos valores y, de acuerdo al análisis realizado en las ecuaciones 2.3, se pueden calcular la ganancia y la frecuencia de corte de este bloque, estos resultados se presentan en las ecuaciones 4.4.

$$G = \frac{R4}{R5} = 20 = 26 [dB] \quad (4.4)$$

$$f_c = \frac{1}{2\pi R4 \cdot C3} = 159 [kHz]$$

Con lo presentado anteriormente, se obtienen los resultados presentados en la figura 4.7. En ésta se puede apreciar que la ganancia se comporta como lo esperado para bajas frecuencias, y se tiene una frecuencia de corte alrededor de los 159 [kHz].

Simulación Circuito Analógico

Para concluir con la sección de simulaciones, se presenta a continuación la simulación de toda la sección analógica, incluyendo la etapa de comparación. En primer lugar, el esquemático utilizado se presenta en la figura 4.8. En ella se observan las etapas de amplificación simuladas anteriormente acopladas al arreglo de comparadores.

Para entender la salida de la simulación total, es necesario entender las salidas de las distintas etapas. En primer lugar está la salida del amplificador de carga, la cual ya fue presentada en la figura 4.5. En ella se ve un pulso de voltaje de 44 [mV] de amplitud con un

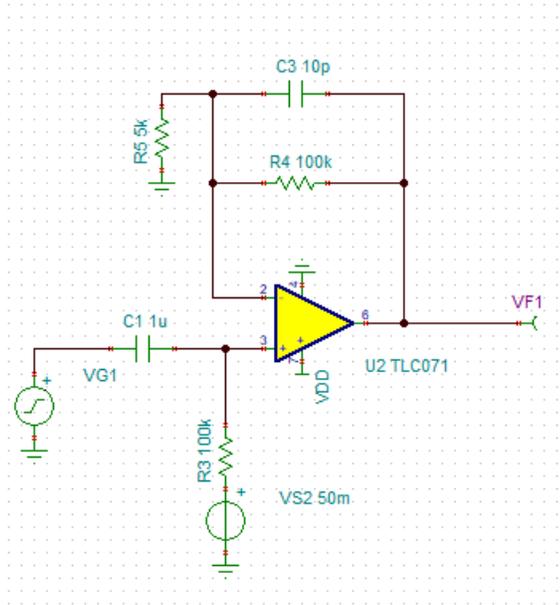


Figura 4.6: Esquemático de la simulación del amplificador no inversor.

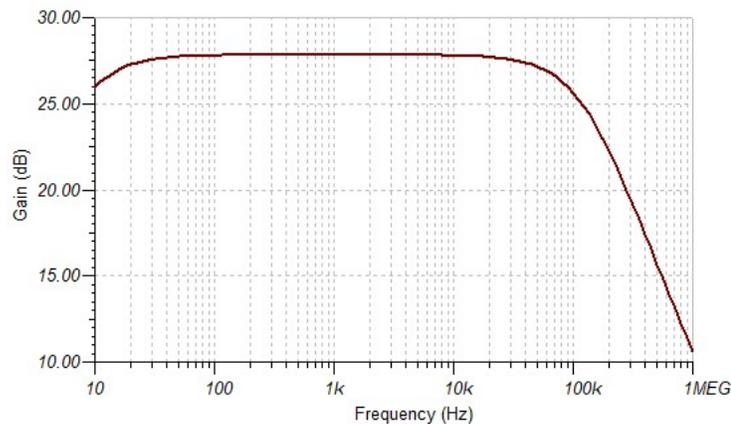


Figura 4.7: Simulación Amplificador No Inversor.

offset de 500 [mV]. Luego, esta señal entra al amplificador de no inversor, el cual tiene como salida la señal presentada en la figura 4.9. En la figura se observa que la amplitud del pulso es de 720 [mV]. El resultado esperado corresponde a $44[mV] \cdot 20 = 880[mV]$, por lo cual, el valor obtenido es aceptable pero no igual.

La señal que entra al arreglo de comparadores es la presentada en la figura 4.9. Este corresponde a un pulso de voltaje de 720 [mV] de amplitud. El propósito de cada comparador es entregar en su salida un pulso cuadrado en caso de que se supere el nivel de voltaje asociado a ese comparador. En particular, los comparadores están configurados con los siguientes voltajes de referencia: 100 [mV], 400 [mV], 1 [V] y 4 [V]. Con esto, se tiene que la salida del sistema total es la presentada en la figura 4.10.

Como era de esperarse, los comparadores con voltajes de referencia igual a 100 [mV] y 400 [mV] entregan un pulso cuadrado en la salida y, por otro lado, los comparadores con voltajes

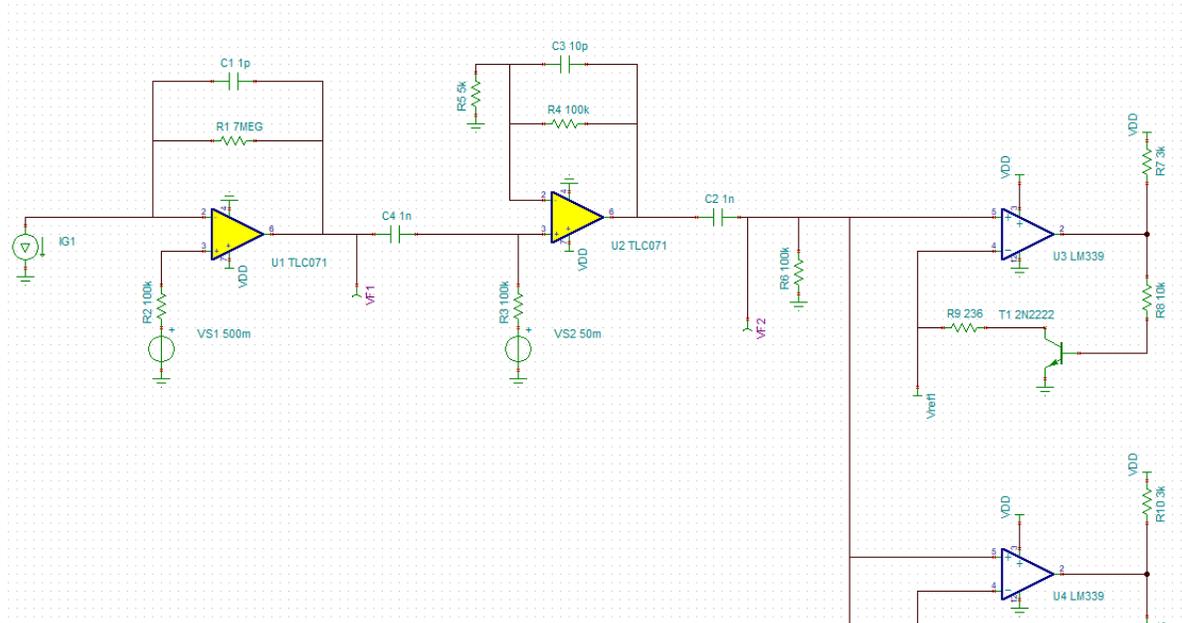


Figura 4.8: Esquemático de la simulación del circuito analógico.

de referencia igual a 1 [V] y 2 [V] se mantiene en 0. Esto se debe a que el pulso es de 720 [mV]. En otras palabras, y tomando como referencia la figura 4.2, la señal que entra a los comparadores supera los niveles de los comparadores 1 y 2, pero no supera los niveles de los comparadores 3 y 4.

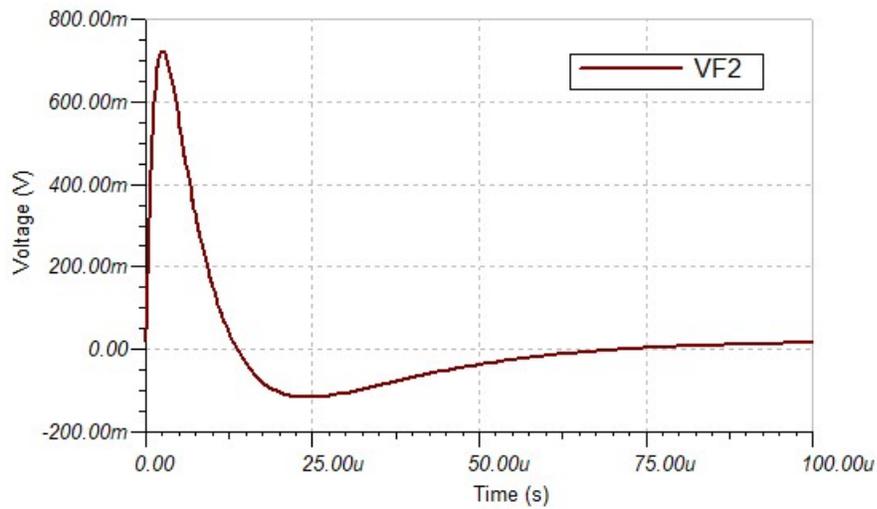


Figura 4.9: Salida del amplificador no inversor.

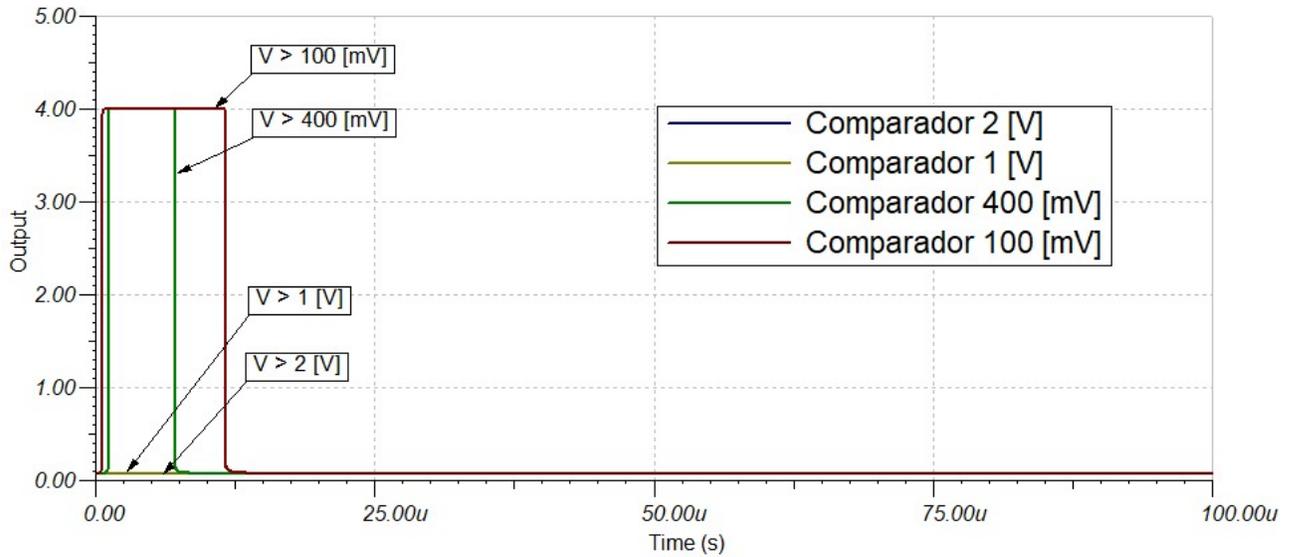


Figura 4.10: Salida comparadores.

Los niveles de voltajes escogidos tienen asociado una energía, es decir, las partículas que superan un nivel, significa que tienen una energía mayor a ese nivel. La energía asociada a cada nivel puede ser calculada si se quitan las ganancias hasta llegar a la carga de entrada. Por ejemplo, para calcular el nivel de energía asociado al nivel de voltaje en primer lugar, se debe quitar la ganancia del amplificador de carga. Esto se hace en la ecuación 4.5.

$$\frac{100 \text{ [mV]}}{20} = 5 \text{ [mV]} \quad (4.5)$$

Luego se debe retroceder un paso más, eliminando la ganancia del amplificador de carga. Esto se realiza en la ecuación 4.6

$$\frac{0,005[V]}{\frac{1}{10^{-12}} [C]} = 5 \times 10^{-15} [C] \quad (4.6)$$

Calculada la carga en 4.6, se puede determinar la energía necesaria para generar esa carga, ya que se sabe que se necesitan 3.6 [eV] para generar una carga de $1,6 \times 10^{-19}$ [9]. Con esto se calcula la energía en la ecuación 4.7

$$E = \frac{3,6 \text{ [eV]} \cdot 5 \times 10^{-15} [C]}{1,6 \times 10^{-19} [C]} \quad (4.7)$$

$$E = 112,5 \text{ [keV]}$$

En conclusión, el nivel de voltaje de 100 [mV] está asociado al nivel de energía de 56.25 [keV]. Es decir, si una partícula de una energía mayor a 56.52 [keV] interactúa con el detector, esta generará un pulso de voltaje cuadrado a la salida del comparador 1.

Bajo el mismo procedimiento detallado anteriormente, se pueden calcular los niveles de energía asociados a los otros niveles de voltaje. Estos se presentan en la tabla 4.1.

Tabla 4.1: Nivel de energía asociado a cada nivel de voltaje.

Voltaje [V]	Energía [keV]
0.1	112.5
0.4	450
1	1125
2	2250

La elección de estos niveles es relativamente arbitraria, a excepción del nivel más alto y el nivel más bajo. El nivel de 100 [mV] se escoge debido a que es un nivel lo suficiente mente bajo para captar las partículas de menor energía y estar por sobre el ruido electrónico. Por otro lado, el nivel de 2 [V] se escoge ya que está asociado a partículas de una energía mayor a 2 [MeV], las cuales se encuentran en menor cantidad.

Capítulo 5

Implementación del instrumento

5.1. Construcción del hardware

5.1.1. Placa PCB

Para llevar a cabo la implementación del diseño presentado en el capítulo anterior, se diseñó una placa PCB en el software KiCAD. Este diseño presenta limitaciones de espacio, la primera corresponde a que la placa debe ir en la cara lateral de uno de los cubos que conforma el satélite, esto implica que tiene que tener a lo más 10 [cm] de alto y 10 [cm] de ancho, en particular, la placa diseñada tiene unas dimensiones de 8.26 [cm] de alto por 9.77 [cm] de ancho. Además, la placa cuenta con un sacado rectangular debido a que a ella se acoplan otros elementos del satélite, por lo que en la práctica todo el circuito debe caber en un espacio de 7.62 [cm] de largo y 4.37 [cm] de ancho. Considerando estas limitaciones de espacio, se diseñó la placa presentada en el anexo.

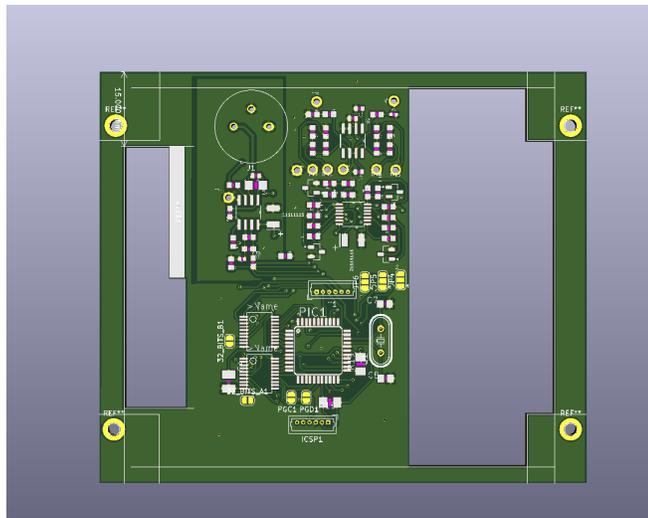


Figura 5.1: Modelo 3D de placa PCB. Tomada de diseño de Cristóbal Garrido.

5.1.2. Etapas del circuito

Para llevar a cabo del diseño presentado, es necesario un circuito que posea cuatro etapas. La primera de ellas es la etapa de amplificación, esta corresponde al circuito ya presentado anteriormente. La segunda de ellas es un circuito que sea capaz de generar los voltajes de referencia para permitir la comparación de los pulsos amplificados por la etapa de amplificación. Luego sigue la etapa de comparación, la cual es la encargada de generar los pulsos de acuerdo a los niveles que supera la señal amplificada. Por último se tiene la etapa de procesamiento digital en donde se cuentan los pulsos y se procesan para ser comunicados a través de I2C.

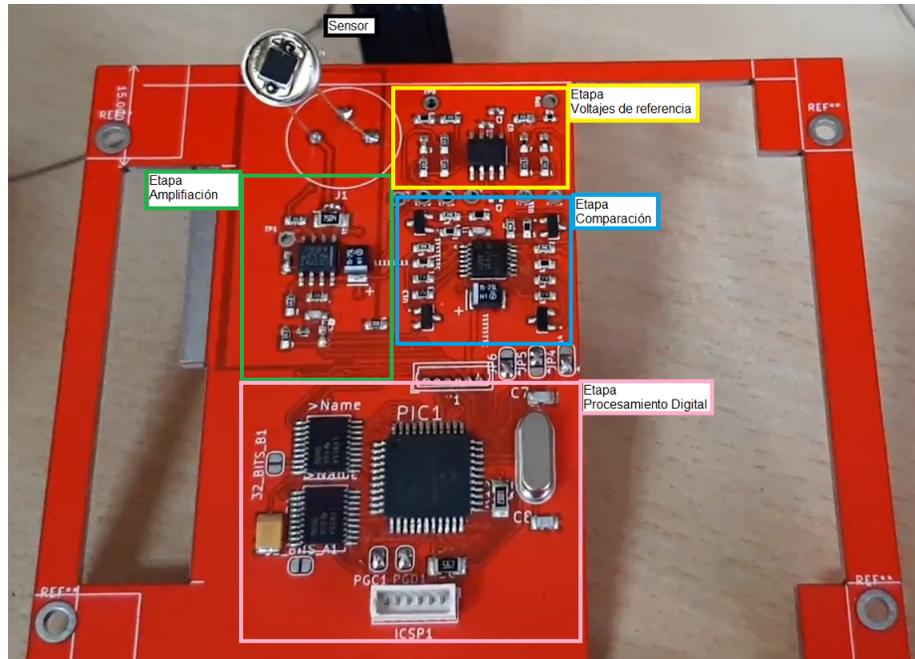


Figura 5.2: Placa del dispositivo con las distintas etapas

Etapa de amplificación

Las componentes para la etapa de amplificación se listan en la tabla presentada en el anexo 8.1, en particular destaca el amplificador operacional TLC072, el cual fue escogida por varias razones. La primera es que el empaquetado cuenta con dos amplificadores operacionales, tal como lo requiere el diseño.

En segundo lugar, es un amplificador de bajo ruido, lo cual es de suma importancia debido a que las señales de entrada son muy pequeñas.

Por último, esta serie de amplificadores presenta un parámetro de *slew rate* muy favorable para la aplicación. El parámetro *slew rate* tiene relación con la velocidad a la cual el amplificador amplifica la señal de entrada. Esto es de suma importancia en esta aplicación ya que las señales de entrada serán pulsos de corriente de una duración muy pequeña, por la tanto es de suma necesidad contar con un amplificador rápido.

Es importante destacar que luego de realizar pruebas con el circuito analógico y digital, se encontró la necesidad de implementar tierras separadas para ambas etapas, en consecuencia,

Tabla 5.1: Parámetros relevantes TLC072

Parámetro	Valor típico
Input Noise Voltage	7 [nV \sqrt{Hz}]
Slew Rate SR-	16 [V/ μs]
Slew Rate SR+	19 [V/ μs]

existe un plano de tierra para la etapa analógica y otro plano de tierra para la etapa digital. Ambas tierras son conectadas a través de una resistencia de 10 [Ω], lo cual genera una diferencia de voltaje de 35 [mV] entre ambos circuitos.

Esta es una práctica común en electrónica debido a que permite evitar el ruido electrónico proveniente de las señales digitales. Las señales digitales consisten en pulsos cuadrados, por lo tanto, generar estos pulsos involucra altos consumos de corriente en tiempos cortos, lo cual produce ruido electrónico.

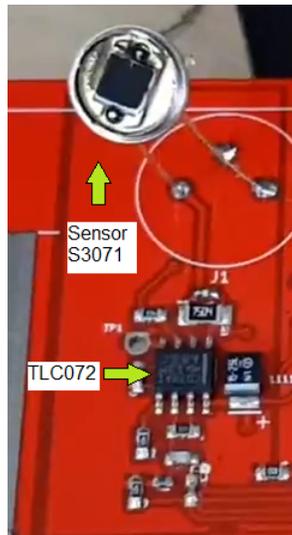


Figura 5.3: Etapa de amplificación

Etapa de voltajes de referencia

Un segundo bloque necesario para llevar a cabo del diseño es un circuito capaz de generar voltajes de referencia para comparar la amplitud de los pulsos de salida de la etapa de amplificación. Además debe generar los voltajes de *offset* presentes a la entrada de los amplificadores operacionales. Al igual que la etapa anterior, la lista completa de las componentes se encuentra en el anexo.

En esta etapa destaca el uso del integrado MAX6126AASA41 el cual es un generador de voltaje de alta precisión y de muy bajo ruido, ambas características son de alta relevancia pues los voltajes a comparar son del orden de los milivoltios y se debe disminuir la mayor cantidad de ruido en el sistema, ya que se está trabajando con señales muy pequeñas y sensores muy sensibles.

El circuito que se implementó funciona de la siguiente manera. Se genera un voltaje fijo de



Figura 5.4: Etapa de voltajes de referencia

Tabla 5.2: Parámetros relevantes MAX6126AASA41

Parámetro	Valor típico
Output Voltage Accuracy	$\pm 0.02\%$
Output Noise Voltage	$1.3 [\mu V_{pp}]$

4.96 [V], este se conecta a un arreglo de resistencias en paralelo que funcionan como divisores de voltaje. De este arreglo se obtienen los voltajes de referencia y los voltajes de *offset*, este arreglo se presenta en la figura 5.5.

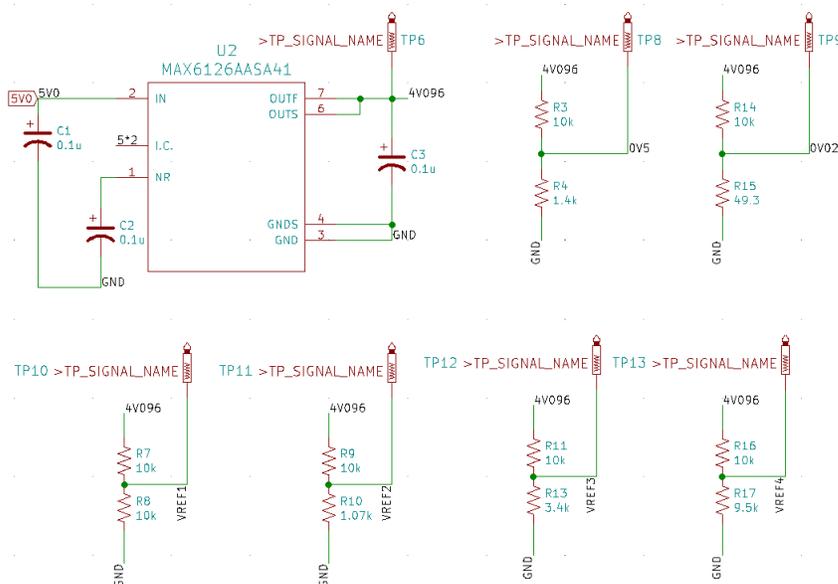


Figura 5.5: Esquemático divisor de voltajes

En la tabla 5.3, se presentan los voltajes de comparación diseñados originalmente en comparación a los voltajes obtenidos en la práctica.

Tabla 5.3: Voltajes de referencia teóricos vs obtenidos

Voltaje de referencia teórico [V]	Voltaje de referencia real [V]
0.1	0.124
0.4	0.395
1	1.002
2	1.995

Etapa de comparación

La tercera etapa del circuito corresponde a la etapa de comparación. En esta etapa se compara la salida de la etapa de amplificación con los voltajes generados en la etapa de voltajes de referencia, en su salida genera pulsos de una amplitud de 5 [V]. La componente más importante de esta etapa es el comparador LMV339IPWR. El empaquetado de este comparador contiene cuatro comparadores y se caracteriza por tener una corriente baja de alimentación.

Cada comparador está asociado a un nivel de voltaje, es decir, cada vez que se supere el nivel de voltaje asociado al comparador, este entregará un pulso en su salida. Con este mecanismo se podrán calcular los niveles de radiación que interactúan con el sensor.

Tabla 5.4: Parámetros relevantes LMV339IPWR

Parámetro	Valor típico
Output Noise Voltage	$\pm 0.02\%$
Output Voltage Accuracy	$1.3 \mu V_{pp}$

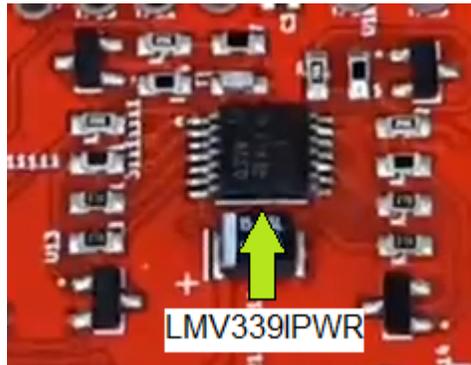


Figura 5.6: Etapa de comparación

Etapa de procesamiento digital

El último bloque que conforma al circuito corresponde a la etapa de procesamiento digital. Esta etapa es la encargada de recibir los pulsos generados por los contadores, registrar cuántos pulsos se generan, almacenar esta información y ser capaz de transmitirla mediante el protocolo de comunicación I2C.

Para implementar este sistema se escogió para realizar las cuentas el integrado SN74LV8154, que corresponde a un contador digital de 16 bits y para almacenar y transmitir los datos se

escogió el microprocesador PIC16F877A. Los detalles sobre el funcionamiento de este sistema se detallarán en la siguiente sección.

El sistema cuenta con dos contadores de 16 bits los cuales en su empaquetado tienen dos contadores cada uno, esto implica que se tiene un total de cuatro contadores, uno por cada comparador de la etapa de comparación.

La elección del microprocesador PIC16F877A se debe principalmente a que posee la cantidad suficiente de pines I/O para realizar la operación deseada y tiene implementado el hardware y software para funcionar como dispositivo esclavo en un sistema de comunicación I2C. En cuanto a la velocidad de operación, este procesador funciona con un cristal de 20 [MHz].

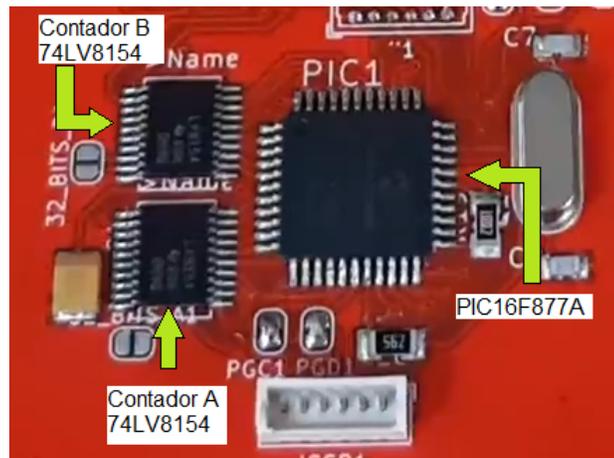


Figura 5.7: Etapa de procesamiento digital

5.2. Procesamiento digital

Para el procesamiento digital de la señal se ocuparán dos herramientas importantes. En primer lugar, se utiliza una Raspberry Pi Zero Wireless, que corresponde a un *Single-Board Computer* capaz de correr un sistema operativo de Linux llamado Raspbian, cuenta con puertos para la comunicación I2C y es una plataforma que ha demostrado ser funcional en otras misiones espaciales, como por ejemplo, satélites de la empresa Surrey [17]. Además, este mini-computador es utilizado en otras misiones del satélite. Por esta razón era un requisito desarrollar un sistema que se pudiera comunicar con la Raspberry Pi.

5.2.1. Contador I2C

El *Contador I2C* es un dispositivo diseñado por Cristián Lopez de la USACH, el cual tiene como función enviar el registro de pulsos cuadrados acumulados durante un tiempo definido en dos contadores de 16 bits. La consulta de registros se realiza mediante el protocolo de comunicación I2C.

La arquitectura consta de los siguientes componentes:

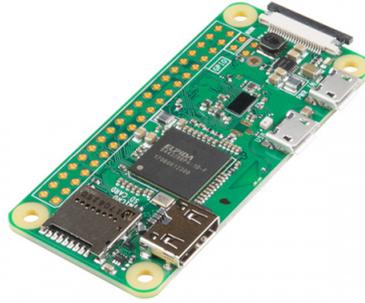


Figura 5.8: Raspberry Pi Zero Wireless

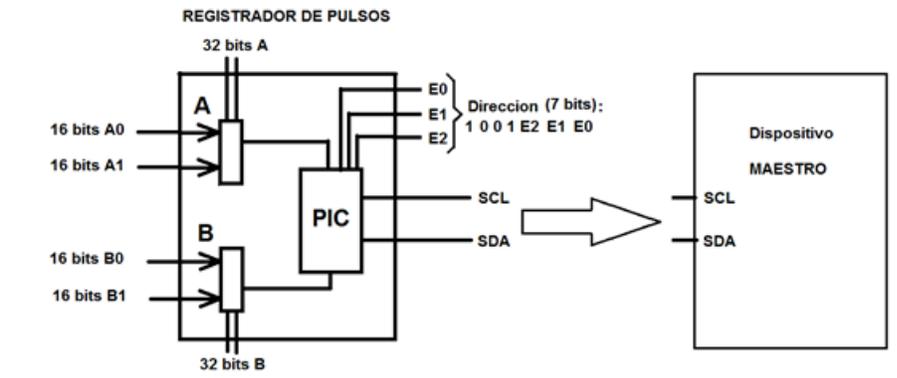


Figura 5.9: Esquema general del Contador I2C. Tomado de Cristián Lopez

- 2 integrados SN74LV8154 (Dual 16-Bit Binary Counters With 3-State Output Registers).
- 1 microcontrolador PIC16F877A.

El integrado SN74LV8154 consta de dos contadores de pulsos de 16 bits. El microcontrolador PIC16F877A, obtiene los registros de los integrados SN74LV8154, enviando por I2C las cuentas acumuladas a petición del dispositivo maestro.

Algoritmo

El algoritmo para obtener las cuentas de los registros de los contadores es el que se presenta en el diagrama de flujo de la figura 5.10. Básicamente, consiste en el algoritmo I2C tradicional, el dispositivo maestro envía la dirección de inicio START, posteriormente envía la dirección del dispositivo esclavo más un bit de escritura, luego se escribe el contador sobre el cual se desean obtener las cuentas, en este punto se pueden solicitar las cuentas sólo del contador A, sólo del contador B o ambas cuentas juntas. Para esta aplicación, siempre se solicitarán las cuentas de ambos contadores, por lo cual el dispositivo maestro debe escribir el número hexadecimal 0xAB, indicando que solicita la cuenta de ambos contadores.

Una vez que el dispositivo esclavo recibe esta petición, procede a mandar los 8 bytes de datos correspondiente al registro de cuentas que llevan los contadores hasta ese minuto. Los

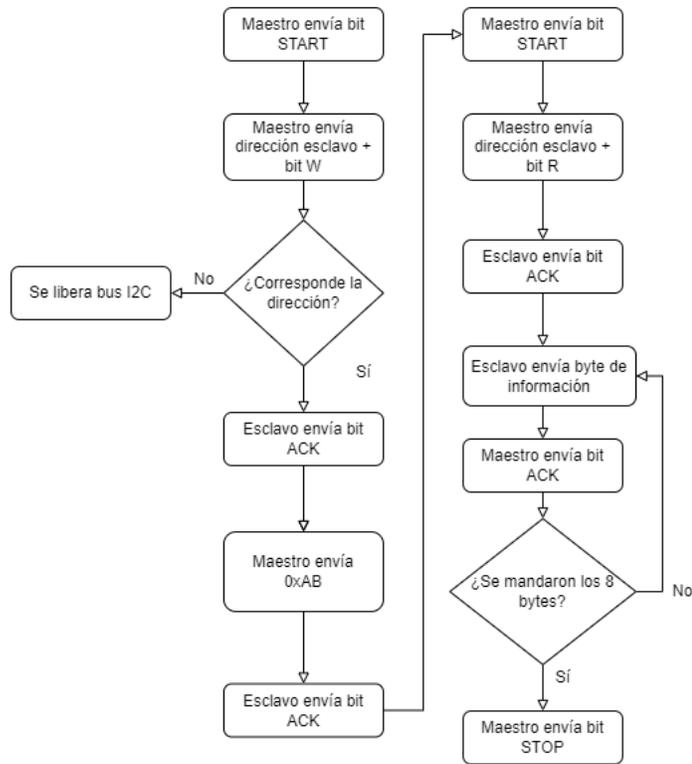


Figura 5.10: Diagrama de flujo del Contador I2C

8 bytes son producto de que se tienen cuatro contadores de 16 bits, esto implica que se requieren 32 bits de información para obtener la cuenta de los cuatro contadores lo cual es equivalente a 8 bytes. En concreto los bytes enviados corresponden a:

$$\begin{aligned}
 \text{Byte 0} &= \text{Byte bajo del registro de cuentas A0} = A0_L \\
 \text{Byte 1} &= \text{Byte alto del registro de cuentas A0} = A0_H \\
 \text{Byte 2} &= \text{Byte bajo del registro de cuentas A1} = A1_L \\
 \text{Byte 3} &= \text{Byte alto del registro de cuentas A1} = A1_H \\
 \text{Byte 4} &= \text{Byte bajo del registro de cuentas B0} = B0_L \\
 \text{Byte 5} &= \text{Byte alto del registro de cuentas B0} = B0_H \\
 \text{Byte 6} &= \text{Byte bajo del registro de cuentas B1} = B1_L \\
 \text{Byte 7} &= \text{Byte alto del registro de cuentas B1} = B1_H
 \end{aligned}
 \tag{5.1}$$

Cada contador posee un registro bajo y alto, el registro bajo corresponde a los 8 bits menos significativos del valor de la cuenta y el registro alto corresponde a los 8 bits más significativos, por lo tanto, para obtener el valor de cada contador se realiza la siguiente operación en términos de bits:

$$\begin{aligned}
\text{Contador A0} &= 256 \cdot A0_H + A0_L \\
\text{Contador A1} &= 256 \cdot A1_H + A1_L \\
\text{Contador B0} &= 256 \cdot B0_H + B0_L \\
\text{Contador B1} &= 256 \cdot B1_H + B1_L
\end{aligned}
\tag{5.2}$$

Finalmente, una vez de que el dispositivo maestro recibió los 8 bytes de los registros de los contadores, éste envía la señal STOP con lo cual termina la comunicación.

Otra forma de visualización de la comunicación I2C se adjunta en el anexo 8.3. En ella se muestra un diagrama similar al presentado en la figura 2.10 pero con la trama I2C del algoritmo explicado en esta sección.

Dirección del dispositivo

En cuanto a la dirección del dispositivo, esta es definida en hardware a través de los jumpers $J5$, $J6$ y $J7$ dispuestos en el circuito, estos se observan en el esquemático del contador presentado en el anexo 8.2, estos jumpers corresponden a los pines E0, E1 y E2 mostrados en la figura 5.9. En particular, se escogió la dirección $1001010 = 0x4A$, esto implica que la configuración de los jumpers se establece como $J7 = 0$ [V] $J6 = 3.3$ [V] $J5 = 0$ [V].

5.2.2. Driver I2C para Raspberry Pi Zero Wireless

Para realizar la comunicación entre el contador y la Raspberry Pi es necesario implementar un driver que permita la comunicación entre el microprocesador PIC16F877A y la Raspberry Pi Zero Wireless. Este fue implementado en *Python* utilizando la librería SMBus.

La librería permite un manejo sencillo del bus I2C implementado en la Raspberry Pi, a través de sus distintos métodos. Es de particular interés encontrar un método que realice la secuencia implementada en el código del microcontrolador del contador I2C, es decir, la secuencia de bits y de envío de datos debe ser la misma. La trama I2C del contador se puede observar en el anexo 8.3, con esta información fue necesario encontrar una función que cumpliera con esta secuencia, para ello se consultó a la documentación de la librería y se encontró que la función `read_i2c_block_data(Addr, Comm, Size)` cumple con el requisito, como se presenta en la figura 5.11.

```

S Addr Wr [A] Comm [A]
S Addr Rd [A] [Data] A [Data] A ... A [Data] NA P

```

Figura 5.11: Trama I2C de la función en python

Los parámetros de entrada de esta función corresponden a la dirección del dispositivo esclavo `Addr`, un registro desde el cual se leerá la información `Comm` y el tamaño del bloque que se leerá, es decir, la cantidad de bytes a ser transmitidos. En este caso particular y dada la implementación en el software del microprocesador, la dirección corresponde a $0x4A$ como

se mencionó anteriormente, luego el registro `Comm` equivale a indicar de qué contador se desea leer la información, en general siempre se desea leer ambos contadores por lo cual el valor de entrada sería `0xAB`, por último el parámetro `Size` corresponde a la cantidad de bytes a ser transmitidos que en este caso corresponde a 8 bytes.

Aplicando este método con las entradas mencionadas anteriormente, se obtienen un arreglo de largo 8 con los bytes de los registros de los contadores. A partir de esta información se obtiene la cuenta acumulada en cada contador.

El código completo se puede encontrar en el anexo 8.1.

Clock Stretching y Bug Raspberry Pi

Una de las complicaciones que se encontró al momento de implementar el driver que permite la comunicación entre la Raspberry Pi Zero W y el microprocesador PIC16F877A a través de I2C fue que se presentaba el error [Errno 121] Remote I/O error. Este error implica de que las conexiones del bus I2C no están conectadas correctamente o que hay un problema de sincronización con el reloj.

Como se mencionó anteriormente, la Raspberry Pi funciona como el dispositivo maestro del sistema de comunicación, por lo tanto, es el dispositivo que define la velocidad de la transmisión de datos a través de la velocidad de la línea SCL. Una de las primeras soluciones que se implementó fue modificar la velocidad de transmisión de 100 [kbits/s] a 400 [kbits/s]. El problema de esta solución es que el contador no es el único dispositivo esclavo del sistema, por lo cual era necesario que operara a la velocidad por defecto de 100 [kbits/s].

Luego de investigar en diversos foros y analizar el código del esclavo en profundidad se encontró que el problema era el siguiente: en la línea 88 del código del esclavo presentado en el anexo 8.1, la línea original corresponde a lo presentado en el código 5.1

Código 5.1: Línea 88 código del esclavo

```
1 SSPCON.CKP = 0;
```

De acuerdo al manual del microprocesador PIC16F788A, en el modo esclavo I2C, el bit CKP del registro SSPCON1 al tener el valor 0, implica que mantiene la línea SCL en LOW de tal manera de asegurar la transferencia de datos, esto implica que el dispositivo está aplicando *clock stretching*.

Por otro lado, se realizaron pruebas en distintos modelos de la Raspberry Pi, estos fueron los siguientes: Raspberry Pi Zero W, Raspberry Pi Zero W 2, Raspberry Pi 3 y Raspberry Pi 4. De las pruebas realizadas el programa funciona a 100 [kbits/s] sólo en la Raspberry Pi 4.

Revisando la documentación, se encontró que los modelos de Raspberry Pi con SoC (*System on Chip*) distinto al Broadcom BCM2711, es decir, todos los modelos distintos a la Raspberry Pi 4 [18], presentan un bug [19] de implementación en su implementación de I2C que puede producir una corrupción en la transmisión de datos. Esto implica que no se pueden conectar dispositivos que implementen *clock stretching*. En particular, la Raspberry Pi Zero

Tabla 5.5: SoC de los distintos modelos de Raspberry

Modelo	SoC
Raspberry Pi Zero W	BCM2835
Raspberry Pi Zero 2 W	BCM2710A1
Raspberry Pi 3	BCM2837B0
Raspberry Pi 4	BCM2711

W que utiliza el procesador Broadcom BCM2835. En la tabla 5.5, se pueden ver los SoC de las distintas Raspberry Pi.

Como se mencionó anteriormente, era imperativo el uso de la Raspberry Pi Zero W, primeramente por sus dimensiones reducidas y, en segundo lugar, porque otras misiones del satélite la utilizaban. Por lo tanto, la solución fue deshabilitar *clock stretching* en el código del esclavo, lo cual en la práctica se traduce a mantener el bit CKP del registro SSPCON1 en 1, tal como se presenta en el código final del esclavo.

Capítulo 6

Pruebas realizadas y resultados

En el siguiente capítulo se presentan las pruebas realizadas al dispositivo. Se realizaron pruebas en dos condiciones diferentes, la primera fue con el dispositivo sin el sensor, y la segunda con el dispositivo con el sensor incluido. Sin embargo, sólo se presentan las pruebas realizadas sin sensor ya que, al conectar el sensor, no se detectan mediciones por lo cual no se registran pulsos en la salida de la etapa de amplificación.

En ambas condiciones, las conexiones para realizar las pruebas fueron las presentadas en el diagrama de la figura 6.1. Se alimenta la placa utilizando los pines de 5 [V], 3.3 [V] y GND de la Raspberry Pi Zero, para la comunicación se conectan los pines SDA y SCL de ambos dispositivos.

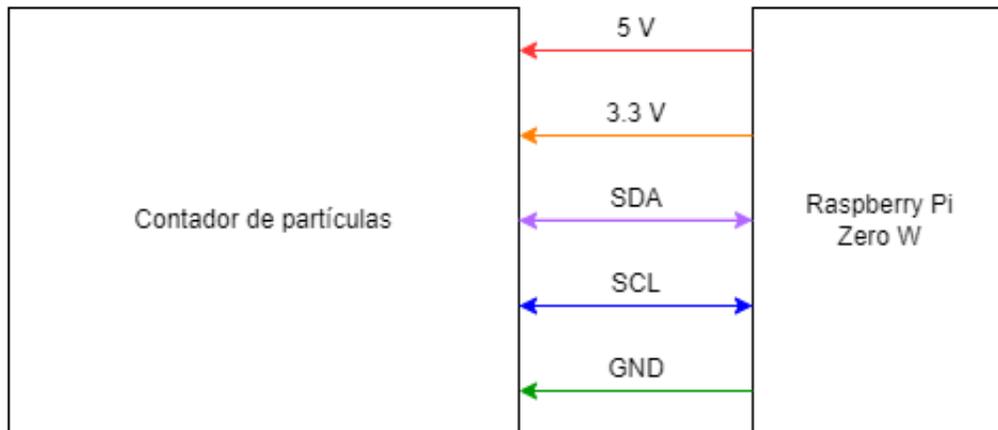


Figura 6.1: Diagrama de conexiones para pruebas

6.1. Pruebas sin el sensor

Para las pruebas realizadas sin el sensor, se realizan las conexiones presentadas en la figura 6.1. La configuración en la práctica se puede observar en la imagen presentada en la figura 6.2. Es importante destacar que los pulsos pedidos en esta sección son sólo ruido. Esto se debe a que la entrada del amplificador de carga es de alta impedancia, por lo tanto, al estar

desconectada genera altos niveles de ruido.

Lo anterior implica que los resultados mostrados en la sección, sólo permiten comprobar de que la electrónica está funcionando, sin embargo, no se pueden obtener conclusiones sobre si el sistema detecta correctamente la radiación.

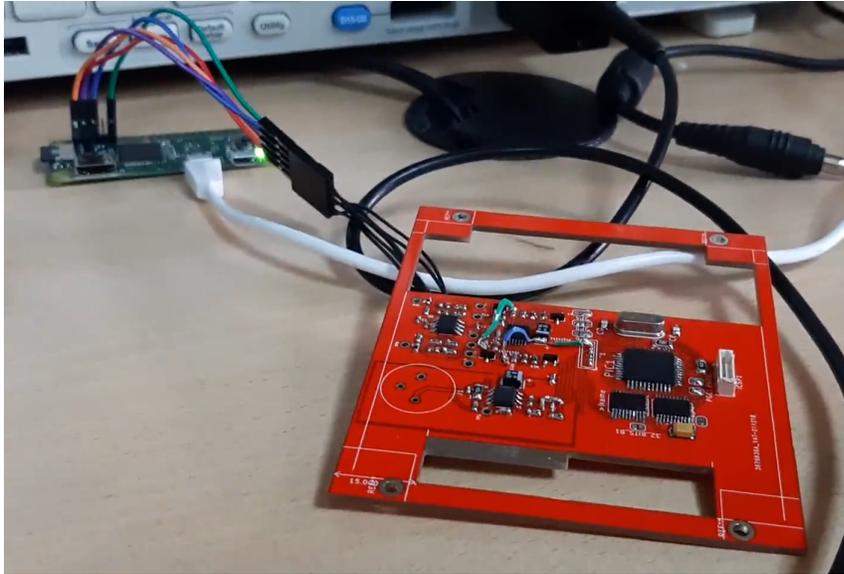


Figura 6.2: Configuración de pruebas sin sensor

6.1.1. Resultados etapa amplificación

En primer lugar, se presentan los resultados obtenidos para la salida de la etapa de amplificación. Se presenta la salida del amplificador en dos momentos diferentes ya que se considera pertinente visualizar la señal en dos escalas diferentes, en particular, la escala más pequeña es con el osciloscopio en una resolución de 50 [mV] y 100 [ms] (figura 6.3) y la escala más grande tiene una resolución de 500 [mV] y 100 [ms] (figura 6.4) .

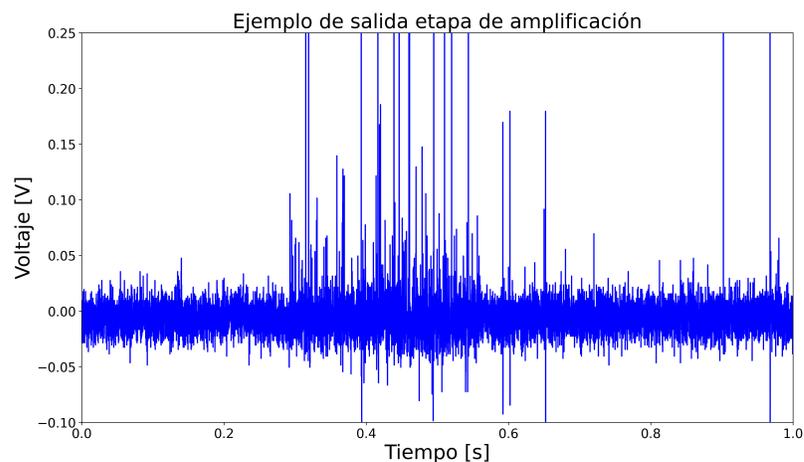


Figura 6.3: Salida etapa de amplificación escala 50 [mV]

Estos resultados muestran ráfagas de pulsos que eran medidos por el sistema, por esta razón no existe una consistencia entre ambas figuras. Cabe destacar que estas señales fueron

obtenidas utilizando como referencia la tierra de la etapa de amplificación, por lo tanto, estas señales tienen un offset con respecto al circuito digital el cual corresponde a 35 [mV]. Considerando esto y la tabla 5.3, los valores que se consideraran válidos para los comparadores serán los presentados en la tabla 6.1.

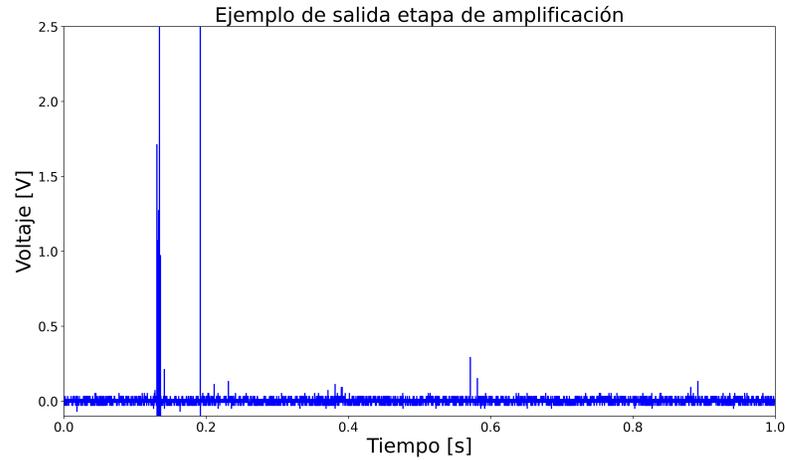


Figura 6.4: Salida etapa de amplificación escala 500 [mV]

Tabla 6.1: Voltajes de referencia teóricos vs obtenidos vs válidos

Voltaje de referencia teórico [V]	Voltaje de referencia real [V]	Voltaje válido [V]
0.1	0.124	0.089
0.4	0.395	0.36
1	1.002	0.967
2	1.995	1.96

Con estas consideraciones se puede observar en la figura 6.3, hay una gran cantidad de pulsos que superan el nivel de 89 [mV], en particular para este caso son 130. Por otra parte, al observar la salida del amplificador en una escala mayor, como se muestra en la figura 6.4, se pueden ver los pulsos que superan los otros niveles de voltaje, en este caso 7 superan el nivel de 360 [mV] como también el de 967 [mV] y por último, sólo 2 superan el nivel de 1.96 [V].

6.1.2. Resultados etapa comparación

En esta sección se presentan los resultados obtenidos luego de la etapa de comparación. Cabe destacar que estos resultados no son producto de las señales presentadas en la sección anterior ni tampoco son de una misma señal para cada contador, es decir, cada tren de pulsos presentado corresponde a un conteo de una señal diferente. A pesar de lo anterior, estos datos sí fueron obtenidos bajo las condiciones descritas al inicio de este capítulo.

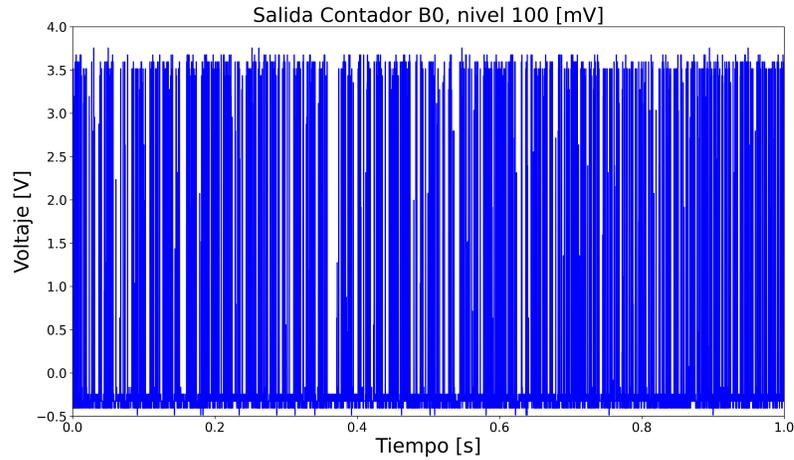


Figura 6.5: Salida comparador conectado a la entrada B0, nivel 100 [mV]

Los pulsos obtenidos en la figura 6.5 corresponden los pulsos de salida del comparador B0 correspondiente al nivel de 100 [mV]. La cantidad de pulsos que superaron los 100 [mV] en esa ventana de tiempo fueron 739 pulsos.

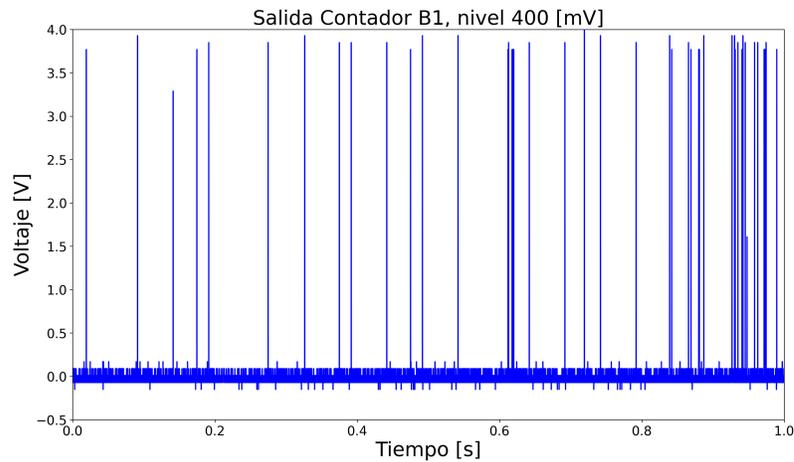


Figura 6.6: Salida comparador conectado a la entrada B1, nivel 400 [mV]

Luego los pulsos obtenidos en la figura 6.6 corresponden los pulsos de salida del comparador B1 correspondiente al nivel de 400 [mV]. La cantidad de pulsos que superaron los 400 [mV] en esa ventana de tiempo fueron 44 pulsos.

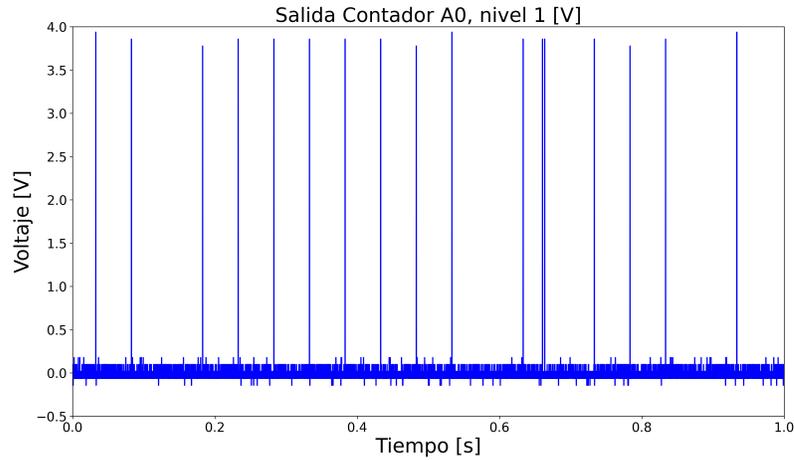


Figura 6.7: Salida comparador conectado a la entrada A0, nivel 1 [V]

En cuanto los pulsos obtenidos en la figura 6.6 corresponden los pulsos de salida del comparador A0 correspondiente al nivel de 1 [V]. La cantidad de pulsos que superaron 1 [V] en esa ventana de tiempo fueron 17 pulsos.

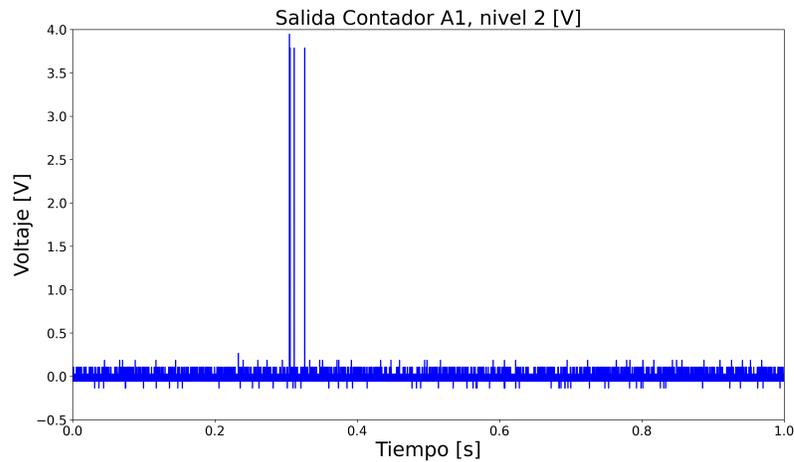


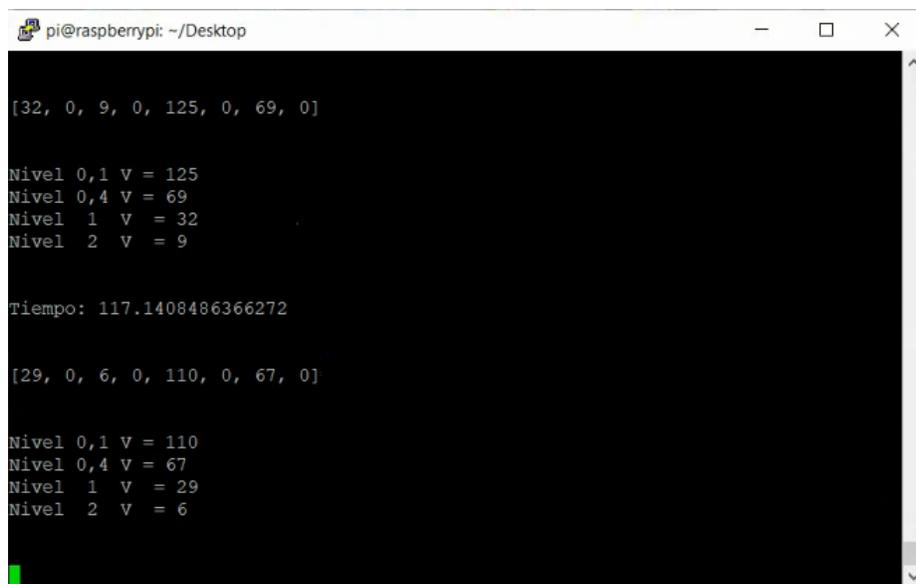
Figura 6.8: Salida comparador conectado a la entrada A1, nivel 2 [V]

Por último, los pulsos obtenidos en la figura 6.6 corresponden los pulsos de salida del comparador A1 correspondiente al nivel de 2 [V]. La cantidad de pulsos que superaron 2 [V] en esa ventana de tiempo fueron 5 pulsos.

6.1.3. Resultados procesamiento digital

Para obtener los resultados de la etapa de procesamiento digital se utilizó la siguiente configuración. En primer lugar se utilizó el software PuTTY para conectarse a través de SSH a la Raspberry Pi desde un computador personal. Luego, se ejecutó el código 8.1 correspondiente al driver I2C presentado en el anexo.

En la imagen presentada en la figura 6.9 se observa una señal similar a la señal de las figuras 6.3 y 6.4, en el sentido de que la cantidad de pulsos que superan los distintos niveles son del mismo orden.



```
pi@raspberrypi: ~/Desktop

[32, 0, 9, 0, 125, 0, 69, 0]

Nivel 0,1 v = 125
Nivel 0,4 v = 69
Nivel 1 v = 32
Nivel 2 v = 9

Tiempo: 117.1408486366272

[29, 0, 6, 0, 110, 0, 67, 0]

Nivel 0,1 v = 110
Nivel 0,4 v = 67
Nivel 1 v = 29
Nivel 2 v = 6
```

Figura 6.9: Resultados etapa de procesamiento digital

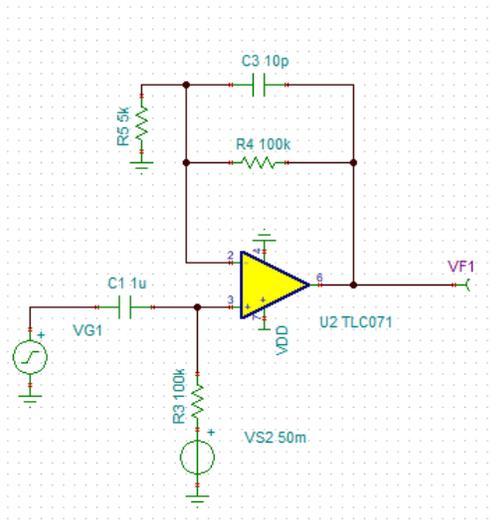
Capítulo 7

Discusión y Análisis

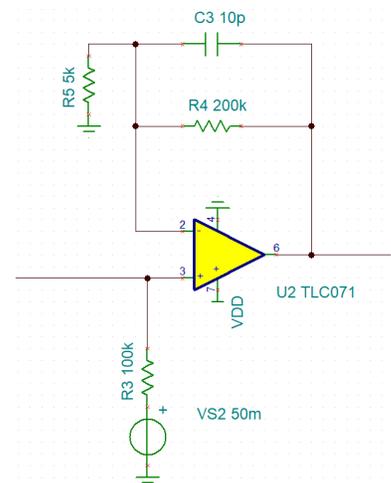
En la siguiente sección se discutirán las diferencias entre la simulación y la implementación real, los resultados obtenidos, también tareas pendientes que se necesitan para concluir el trabajo y cómo adaptar el detector construido para implementar la medición de la radiación gamma.

7.1. Simulación vs Implementación Real

La principal diferencia con la simulación fue que se cambió la ganancia del amplificador no inversor. Como se puede ver en los esquemáticos de la figura 7.1, originalmente, se tenía una resistencia de $100 [k\Omega]$ en el loop de retroalimentación del amplificador no inversor, en cambio en la implementación real se escogió una resistencia de $200 [k\Omega]$.



(a) Amplificador No Inversor de la simulación



(b) Amplificador No Inversor implementado

Figura 7.1: Cambio Amplificador No Inversor

Este cambio tiene implicancias en la ganancia del amplificador. Recordando las fórmulas

presentadas en 2.3, se tiene que la ganancia del amplificador no inversor no implementado se calcula según lo presentado en la ecuación 7.1.

$$G = \frac{R4}{R5} = 40 = 32 \text{ [dB]} \quad (7.1)$$

Este cambio no produce un cambio significativo en el funcionamiento general del circuito, pero si significa un cambio al momento de calcular la cantidad de radiación, ya que ahora los niveles de voltajes están asociados a otros niveles de energía. Los niveles de energía originales y el cálculo de estos se presentan en la tabla 4.1.

Dado que el cálculo de la energía se realiza mediante operaciones lineales, tal como se muestra en las ecuaciones 4.5, 4.6 y 4.7. Basta dividir por 2 los valores de la tabla 4.1 y se obtienen los nuevos niveles de energía para los distintos niveles de voltaje. Estos se presentan en la tabla 7.1

Tabla 7.1: Nivel de energía asociado a cada nivel de voltaje.

Voltaje [V]	Energía [keV]
0.1	56.25
0.4	225
1	562.5
2	1125

7.2. Análisis de los resultados

Como se mencionó en el capítulo anterior en la sección 6.1, los resultados presentados se obtuvieron sin el sensor. Dado que el amplificador de carga es de alta impedancia, significa que al dejar la entrada desconectada genera altos niveles de ruido. Por lo tanto no se puede concluir que el detector cumple su objetivo principal, pero sí se puede observar el comportamiento de la electrónica.

En cuanto a la etapa de amplificación, se puede observar en las figuras 6.3 y 6.4 el ruido amplificado a la salida de esta etapa, es decir, a la salida del amplificador no inversor. Estas señales son un ejemplo de lo que entra a la etapa de comparación, a modo de referencia, para entender los resultados de los contadores. Se puede observar que los peaks llegan hasta los 2.5 [V].

En cuanto a los resultados de la etapa de comparación, se puede observar en las figuras 6.5, 6.6, 6.7 y 6.8 que los comparadores asociados a las entradas B0, B1, A0 y A1, están generando pulsos de manera correcta. Sin embargo no se puede confirmar de que la cantidad de pulsos en la ventana de tiempo se condice con la señal de salida de la etapa de amplificación, ya que no fueron medidas al mismo tiempo.

A pesar de lo anterior y pasando a los resultados obtenidos para la etapa de procesamiento digital, se puede decir que tanto el software implementado en el microprocesador

PIC16F877A, como el driver programado en la Raspberry Pi Zero W, están funcionando correctamente y logran comunicarse mediante el protocolo I2C. En cuanto a los resultados obtenidos en pantalla, se puede observar que la cantidad de pulsos para cada nivel difiere de los resultados mostrados en las figuras 6.5, 6.6, 6.7 y 6.8. Sin embargo esto se debe a que fueron tomados en instantes diferentes.

Una similitud que se puede encontrar es que mientras más bajo el nivel de voltaje, existe una mayor cantidad de pulsos registrados. Por ejemplo, para el nivel de 100 [mV], en el caso de la figura 6.5 se registran 739 pulsos en un segundo. Por otra parte, en el caso de los resultados de la etapa digital, en la figura 6.9, en el nivel de 100 [mV] se registran 110 pulsos. Análogamente, para el nivel de 2 [V], en el caso de la figura 6.8 se registran 5 pulsos y en la figura 6.9 se registran 6 pulsos para el mismo nivel.

7.3. Trabajo Futuro

Para llegar a resultados concretos en cuanto al funcionamiento del detector, es necesario realizar una evaluación del funcionamiento de este. Para eso se propone en esta sección el trabajo a futuro necesario para lograr este objetivo.

7.3.1. Pruebas con fuente de radiación

Luego de las pruebas realizadas con el generador de funciones, se propone realizar pruebas con una fuente de radioactiva. Para esto se propone contactar con la Comisión Chilena de Energía Nuclear (CCHEN) del Ministerio de Energía, dado que ya ha trabajado en conjunto con el laboratorio SPEL.

Se propone realizar pruebas para medir desde que punto el detector comienza a registrar cuentas, el punto de saturación y el ángulo de medición.

Para determinar desde qué punto el instrumento comienza a registrar cuentas, se propone ir aumentando la fuente de radiación paulatinamente hasta encontrar este punto. De la misma manera, aumentar la cantidad de radiación que a la que se expone el instrumento hasta encontrar el punto de saturación.

En cuanto al ángulo de radiación, se propone orientar el instrumento vertical y horizontalmente. En ambos casos, la placa debe estar perpendicular al suelo. Luego, activar la fuente radioactiva, girar el detector y analizar cómo varían los resultados.

7.3.2. Detector Gamma

Para adaptar el detector a un detector de radiación gamma se deben realizar cambios sencillos en el dispositivo. La ventaja del diseño es que la electrónica se mantiene igual, sin embargo, se debe cambiar el sensor S3071 por un Hamamatsu S7478.

Por otra parte, la detección funciona a través de un centellador, tal como fue explicado en el marco teórico, este material libera fotones ante la interacción con la radiación gamma. El centellador propuesto es el GAGG de EPIC Crystal, las dimensiones de este son $5 \times 5 \times 5$

Capítulo 8

Conclusión y trabajos futuros

8.1. Conclusión

En cuanto a los objetivos planteados, en el presente trabajo se ha logrado diseñar e implementar electrónica y softwares con el objetivo de detectar la radiación corpuscular, acomodable a un satélite CubeSat 3U o superior. Sin embargo, esta no ha podido ser evaluada con una fuente de radiación. Esta evaluación es necesaria para cumplir el objetivo final y establecer con seguridad que el instrumento mide la radiación.

Se ha diseñado un circuito a través del uso de herramientas de simulación utilizando el software Tina-TI. Luego, utilizando software de diseño tipo EDA, se ha diseñado una placa PCB capaz de soportar el diseño y con las dimensiones necesarias para ser acoplada en un satélite CubeSat.

El diseño del sistema consiste en una etapa analógica capaz de medir la radiación del ambiente, esto mediante la interacción de la partículas de alta energía con un material semiconductor. Este material emite un pulso de corriente al interactuar con la radiación de una amplitud proporcional a la energía de la partícula. Luego es transformado en un pulso de voltaje mediante un amplificador de carga. A través de un arreglo de comparadores, se miden los distintos niveles que este pulso de voltaje supera y se realiza una cuenta en el tiempo sobre cuántos pulsos de voltaje superaron cada nivel. Esta información es registrada por un sistema digital, el cual es capaz de contar los pulsos, almacenarlos y comunicarlos a través del protocolo I2C.

El sistema digital que se ha implementado, está conformado por contadores de pulsos, un procesador PIC16F877A y una Raspberry Pi Zero Wireless. Los contadores son los encargados de registrar las cuentas y almacenarlas temporalmente, luego el microprocesador es el encargado de solicitar las cuentas y poder comunicarla a través del protocolo I2C. Por último la Raspberry Pi Zero Wireless con el microprocesador conforman un sistema de comunicación I2C en donde la Raspberry funciona como el dispositivo maestro y el PIC como el dispositivo esclavo, de esta forma la Raspberry solicita la información a través de I2C y es capaz de almacenarla y procesarla para poder ser enviada al sistema central que controla el satélite y sus cargas.

En cuanto a los resultados obtenidos, estos no permiten concluir que el dispositivo es capaz de medir la radiación, pero sí permiten concluir que la electrónica implementada está operando. Para caracterizar el dispositivo se ha propuesto una serie de mediciones con una fuente radioactiva, éstas permitirían evaluar el instrumento y, de esta manera, determinar si es que efectivamente es capaz de medir la radiación y conocer sus limitaciones.

Además, se hizo una propuesta de cómo adaptar el instrumento para que este sea capaz de detectar la radiación gamma. Esto se puede realizar aprovechando la ventaja de que la electrónica para medir este tipo de radiación es la misma.

Por último, de funcionar correctamente en las pruebas con una fuente radioactiva, este dispositivo permitirá caracterizar el ambiente espacial a un muy bajo costo. Esta información sobre las condiciones a las que se enfrentan los satélites al orbitar el planeta tierra y estar expuestos al espacio exterior, serán de mucha utilidad para futuras misiones, proyectos de tecnologías más avanzadas y la comunidad científica.

Bibliografía

- [1] S. W. Janson, *A Brief History of Nanosatellites*. Wiley Online Library, 2019.
- [2] D. M. Periáñez, “A contribution to the characterization of radiation-induced soft errors in sigma-delta modulators and sram memories.” <http://hdl.handle.net/11201/148866>. Accessed: 2022-8-5.
- [3] B. Jacob, “Killer electrons.” https://www.nasa.gov/vision/universe/solarsystem/killer_electrons.html. Accessed: 2022-03-5.
- [4] A. Kennedy, “Biological effects of space radiation and development of effective counter-measures,” *Life Sciences in Space Research*, vol. 1, pp. 10–43, 04 2014.
- [5] D. LaRocca, P. Kaaret, D. Kirchner, A. Zajczyk, W. Robison, T. Johnson, K. Jahoda, W. Fuelberth, H. Gulick, R. McCurdy, K. White, and D. Miles, “Design and construction of the x-ray instrumentation onboard the halosat cubesat,” *Journal of Astronomical Telescopes, Instruments, and Systems*, vol. 6, p. 1, 03 2020.
- [6] J. Racusin, J. Perkins, M. Briggs, G. de Nolfo, J. Krizmanic, R. Caputo, J. McEnery, P. Shawhan, D. Morris, V. Connaughton, D. Kocevski, C. Wilson-Hodge, M. Hui, L. Mitchell, and S. McBreen, “Burstcube: A cubesat for gravitational wave counterparts,” 08 2017.
- [7] S. Waydo, D. Henry, and M. Campbell, “Cubesat design for leo-based earth science missions,” *IEEE Aerospace Conference Proceedings*, vol. 1, 09 2002.
- [8] P. Kangaslahti, E. Schlecht, J. Jiang, W. Deal, A. Zamora, K. Leong, S. Reising, X. Bosch, and M. Ogut, “Cubesat scale receivers for measurement of ice in clouds,” pp. 42–47, 04 2016.
- [9] G. F. Knoll, *Radiation Detection and Measurment*. Wiley, 2010.
- [10] M. A. M. Ackermann and L. Baldini, “Fermi observations of high-energy gamma-ray emission from grb 080916c,” vol. 323, pp. 1688–1693, 2009.
- [11] C. T. Array, “Rayos gamma.” https://www.cta-observatory.org/wp-content/uploads/2020/07/HojadeEjercicios_RayosGamma.pdf. Accessed: 2022-27-4.
- [12] C. Grupen and I. Buvat, *Handbook of Particle Detection and Imaging*. Springer Science

& Business, 2012.

- [13] NXP, “I2c-bus.” <https://web.archive.org/web/20150422060439/http://www.i2c-bus.org/i2c-bus>. Accessed: 2022-8-5.
- [14] K. T. Jayant Mankar, Chaitali Darode, “Review of i2c protocol,” vol. 2, 2014.
- [15] NXP, “Um10204, i2c-bus specification and user manual.” <https://www.nxp.com/docs/en/user-guide/UM10204.pdf>. Accessed: 2022-8-5.
- [16] F. Arneodo, A. Di Giovanni, and P. Marpu, “A review of requirements for gamma radiation detection in space using cubesats,” *Applied Sciences*, vol. 11, p. 2659, 03 2021.
- [17] S. S. T. Ltd, *SSTL releases spectacular Raspberry Pi camera image and video of the Earth*.
- [18] S. Compare, “Raspberrypi models comparison.” <https://socialcompare.com/en/comparison/raspberrypi-models-comparison>. Accessed: 2022-05-5.
- [19] A. Mechatronic, “Raspberry pi i2c clock-stretching bug.” <https://www.advamation.com/knowhow/raspberrypi/rpi-i2c-bug.html>. Accessed: 2022-05-5.

ANEXOS

Anexo A

A1. Esquemáticos del sistema completo

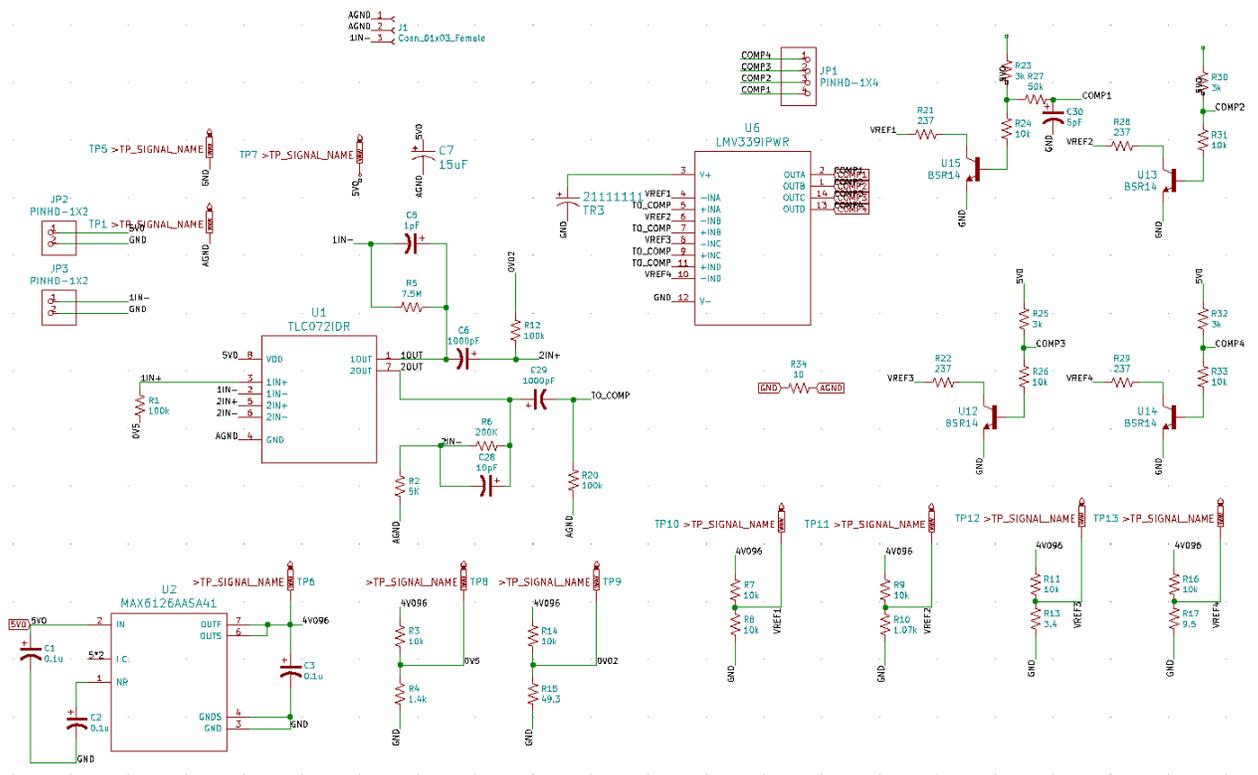


Figura 8.1: Esquemático Circuito Analógico.

Anexo B

B1. Código PIC16F877A

```
1 void funcion_integrado(unsigned short integ);
2 void PWMA(void);
3 void PWMB(void);
4 void cuentas_A(void);
5 void cuentas_B(void);
6 unsigned short bp = 0,aux = 0,ver_dato = 0;
7 unsigned short opc_int[8];
8 unsigned short datos_ent[3];
9 unsigned short f_escritura=0,f_lectura=1;
10 unsigned short direccion_lectura = 0,direccion_escritura = 0,
    direccion_dato = 0, dato = 0,integrado = 0, iniciar_parar = 0;
11 unsigned short aver = 0,interrupcion = 0;
12 unsigned short limpiar;
13 unsigned short aver_que[4];
14 //BITS DE DIRECCION
15 sbit bit_0 at RE0_Bit;
16 sbit bit_1 at RE1_Bit;
17 sbit bit_2 at RE2_Bit;
18 //CONTADOR A(0x0A)
19 sbit GAOL at RC0_Bit;
20 sbit GAOH at RA5_Bit;
21 sbit GA1L at RC5_Bit;
22 sbit GA1H at RC6_Bit;
23 sbit RESET_CONTA at RC7_Bit;
24 //CONTADOR B(0x0B)
25 sbit GBOL at RA0_Bit;
26 sbit GBOH at RA1_Bit;
27 sbit GB1L at RA2_Bit;
28 sbit GB1H at RA3_Bit;
29 sbit RESET_CONTB at RA4_Bit;
30 ///////////////////////////////////////////////////
31 void main() {
32 CMCON = 0x07;
33 ADCON1 = 0x06;
34 TRISA = 0x00;
35 TRISE = 0x07;
36 TRISC = 0x18;
37 TRISD = 0xFF;
38 TRISB = 0xFF;
39 CCP1CON = 0x00;
40 CCP2CON = 0x00;
41 RC1_Bit = 0;
42 RC2_Bit = 0;
43 RESET_CONTA = 1;
44 RESET_CONTB = 1;
45 INTCON = 0b11000000;
46 PIE1.SSPIF = 1;
47 SSPCON = 0b00110110; // Modo esclavo 7 bits de direccion.
48 SSPCON2 = 0x80;
49 SSPSTAT = 0x80;
50 SSPADD = 144 + 8*bit_2 + 4*bit_1 + 2*bit_0;//Direccion del dispositivo
51 T2CON =0b00000100;//Prescaler = 1:1
52 PR2 = 5;
```

```

53 bp= 15;
54 CCP1CON = 0b00001100;
55 CCPR1L = (bp>>2)&0xFF;
56 CCP1CON.CCP1X = bp&0x02;
57 CCP1CON.CCP1Y = bp&0x01;
58 CCP2CON = 0b00001100;
59 CCPR2L = (bp>>2)&0xFF;
60 CCP2CON.CCP2X = bp&0x02;
61 CCP2CON.CCP2Y = bp&0x01;
62     opc_int[0] = 0;
63     opc_int[1] = 0;
64     opc_int[2] = 0;
65     opc_int[3] = 0;
66     opc_int[4] = 0;
67     opc_int[5] = 0;
68     opc_int[6] = 0;
69     opc_int[7] = 0;
70
71 aver_que[0] = 0;
72 aver_que[1] = 1;
73 aver_que[2] = 0;
74 aver_que[3] = 0;
75 direccion_lectura = 0;
76 direccion_escritura = 0;
77 direccion_dato = 0;
78 dato = 0;
79 ver_dato = 0;
80 while(1){
81     //Solo espera interrupcion.
82     }
83 }
84 ////////////////////////////////////////////////////
85 void interrupt(void){
86     interrupcion = 1;
87     if(PIR1.SSPIF){
88         SSPCON.CKP = 1;
89         aver = 1;
90         if((SSPSTAT.R_NOT_W)&(!SSPSTAT.D_NOT_A)){
91             direccion_lectura = SSPBUF;
92             while(SSPSTAT.BF);
93             ver_dato = dato;
94             funcion_integrado(dato);
95             SSPBUF = opc_int[0];
96             while(!SSPSTAT.BF);
97             aver_que[0]++;
98             SSPCON.CKP = 1;
99         }
100         if((SSPSTAT.R_NOT_W)&(SSPSTAT.D_NOT_A)){
101             direccion_dato = SSPBUF;
102             while(SSPSTAT.BF);
103             SSPBUF = opc_int[aver_que[1]];
104             while(!SSPSTAT.BF);
105             aver_que[1]++;
106             if(aver_que[1] == 8){
107                 aver_que[1] = 1;
108             }

```

```

109     SSPCON.CKP = 1;
110     }
111
112     if ((!SSPSTAT.R_NOT_W)&(!SSPSTAT.D_NOT_A)){
113         direccion_escritura = SSPBUF;
114         while(SSPSTAT.BF);
115         aver_que [2]++;
116         SSPCON.CKP = 1;
117     }
118     if(SSPSTAT.D_NOT_A){
119         dato = SSPBUF;
120         while(SSPSTAT.BF);
121         aver_que [3]++;
122         SSPCON.CKP = 1;
123     }
124     PIR1.SSPIF = 0;
125     }
126 }
127 ////////////////////////////////////////////////////
128 void funcion_integrado(unsigned short integ){
129
130     if(integ == 0x0A){
131         cuentas_A();
132         opc_int [4] = 0;
133         opc_int [5] = 0;
134         opc_int [6] = 0;
135         opc_int [7] = 0;
136         RESET_CONTA = 0;
137         RESET_CONTA = 1;
138         PWMA();
139     }
140     if(integ == 0x0B){
141         opc_int [0] = 0;
142         opc_int [1] = 0;
143         opc_int [2] = 0;
144         opc_int [3] = 0;
145         cuentas_B();
146         RESET_CONTB = 0;
147         RESET_CONTB = 1;
148         PWMB();
149     }
150     if(integ == 0xAB){
151         cuentas_A();
152         cuentas_B();
153         RESET_CONTA = 0;
154         RESET_CONTB = 0;
155         RESET_CONTA = 1;
156         RESET_CONTB = 1;
157         PWMA();
158         PWMB();
159     }
160 }
161 ////////////////////////////////////////////////////
162 void cuentas_A(void){
163 CCP1CON = 0x00;
164 if(RC2_Bit){

```

```

165     RC2_Bit = 0;
166     }
167 RC2_Bit = 0;
168 GAOL = 0;
169 GAOH = 1;
170 GA1L = 1;
171 GA1H = 1;
172
173 opc_int[0] = PORTB;//CONTAOL = PORTB;
174
175 GAOL = 1;
176 GAOH = 0;
177 GA1L = 1;
178 GA1H = 1;
179
180 opc_int[1] = PORTB;//CONTAOH = PORTB;
181
182 GAOL = 1;
183 GAOH = 1;
184 GA1L = 0;
185 GA1H = 1;
186
187 opc_int[2] = PORTB;//CONTA1L = PORTB;
188
189 GAOL = 1;
190 GAOH = 1;
191 GA1L = 1;
192 GA1H = 0;
193
194 opc_int[3] = PORTB;//CONTA1H = PORTB;
195 }
196 ////////////////////////////////////////////////////
197 CCP2CON = 0x00;
198 if(RC1_Bit){
199     RC1_Bit = 0;
200 }
201 RC1_Bit = 0;
202 GBOL = 0;
203 GBOH = 1;
204 GB1L = 1;
205 GB1H = 1;
206
207 opc_int[4] = PORTD;//CONTBOL = PORTD;
208
209 GBOL = 1;
210 GBOH = 0;
211 GB1L = 1;
212 GB1H = 1;
213
214 opc_int[5] = PORTD;//CONTBOH = PORTD;
215
216 GBOL = 1;
217 GBOH = 1;
218 GB1L = 0;
219 GB1H = 1;
220

```

```

221 opc_int[6] = PORTD;//CONTB1L = PORTD;
222
223 GBOL = 1;
224 GBOH = 1;
225 GB1L = 1;
226 GB1H = 0;
227
228 opc_int[7] = PORTD;//CONTB1H = PORTD;
229
230 }
231 ///////////////////////////////////////////////////////////////////
232 void PWMA(void){
233 T2CON =0b00000100;//Prescaler = 1:1
234 PR2 = 5;
235 bp = 15;
236 CCP1CON = 0b00001100;
237 CCPR1L = (bp>>2)&0xFF;
238 CCP1CON.CCP1X = bp&0x02;
239 CCP1CON.CCP1Y = bp&0x01;
240 }
241 ///////////////////////////////////////////////////////////////////
242 void PWMB(void){
243 T2CON =0b00000100;//Prescaler = 1:1
244 PR2 = 5;
245 bp = 15;
246 CCP2CON = 0b00001100;
247 CCPR2L = (bp>>2)&0xFF;
248 CCP2CON.CCP2X = bp&0x02;
249 CCP2CON.CCP2Y = bp&0x01;
250 }

```

B2. Código Driver I2C

```
1 from smbus import SMBus
2 import time
3
4 class contador_driver:
5     def __init__(self, i2c_address, bus=1):
6         self.i2cbus = SMBus(bus)
7         self.i2c_address = i2c_address
8
9     def read_measurements(self):
10        data = self.i2cbus.read_i2c_block_data(self.i2c_address, 0xAB,8)
11        cont_a0 = 256*data[1] + data[0]
12        cont_a1 = 256*data[3] + data[2]
13        cont_b0 = 256*data[5] + data[4]
14        cont_b1 = 256*data[7] + data[6]
15
16        print(data)
17        print('\n')
18        print('Contador A0 = '+str(cont_a0))
19        print('Contador A1 = '+str(cont_a1))
20        print('Contador B0 = '+str(cont_b0))
21        print('Contador B1 = '+str(cont_b1))
22        print('\n')
23        return True
24
25
26 def main():
27     cont = contador_driver(0x4A)
28
29     tstart = time.time()
30     tfinish = tstart + 4 * 3600
31
32     while (time.time() <= tfinish):
33         # while True:
34         print('Tiempo: ' + str(time.time() - tstart))
35         print('\n')
36         cont.read_measurements()
37         time.sleep(1)
38
39 if __name__ == "__main__":
40     main()
```

Anexo C

C1. Trama I2C Contador

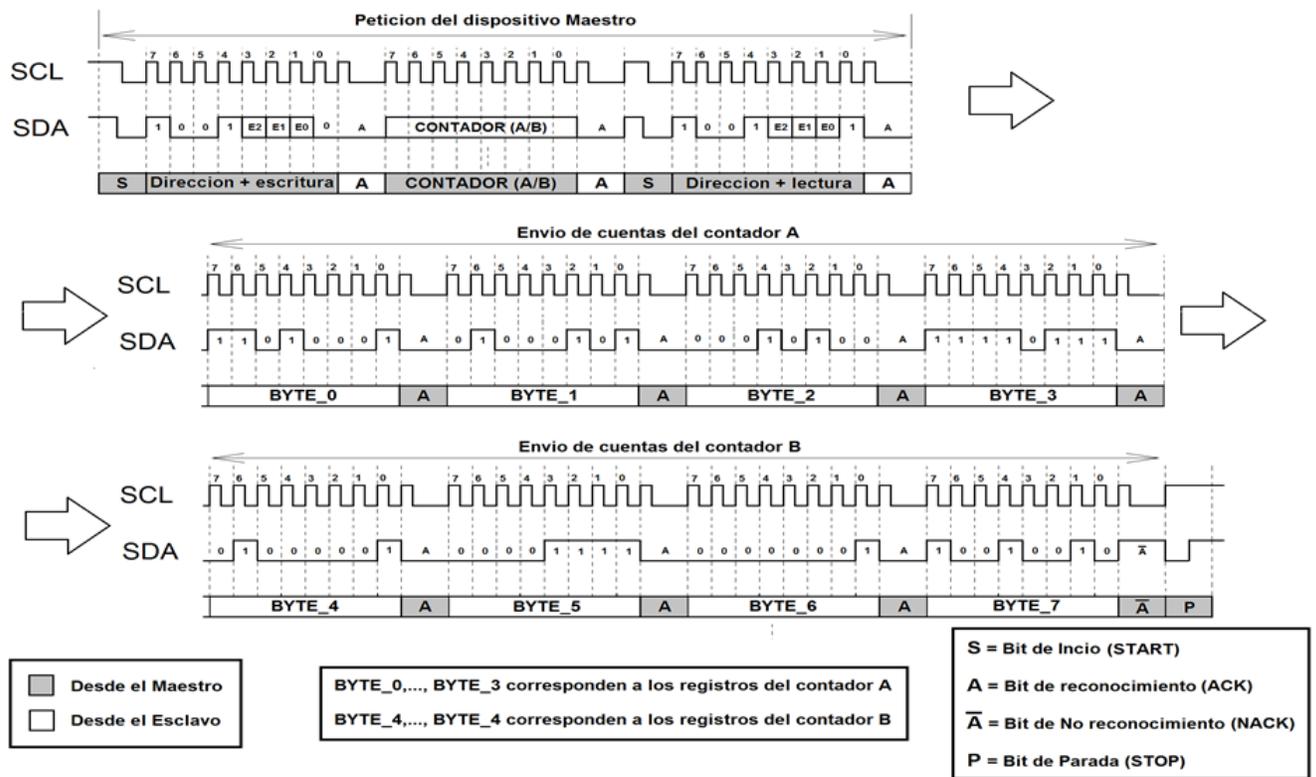


Figura 8.3: Trama I2C Contador. Tomado de Cristián Lopez

Anexo D

D1. Tablas de componentes

Tabla 8.1: Tabla de los componentes del circuito

Componente	Tipo/Valor
C1, C2, C3	15 uF
C4	220 uF
C5	1 pF
C6, C29	1000 pF
C7, C8	22 pF
C28	10 pF
C30	5 pF
CONTA1	74LV8154
CONTB1	74LV8154
PIC1	16F877A
Q1	20 MHz
R1, R12, R20	100k
R2	5K
R4	1.4k
R5	7.5M
R6	200K
R15	49,3
R18	5k6
R19	10k
R21, R22, R28, R29	237
R23, R25, R30, R32	3k
R3, R7-R11, R13, R14, R16, R17, R24, R26, R31, R33	10k
R27	50k
R34	10
U1	TLC072IDR
U2	MAX6126AASA41
U3	MCP9804_MSOP
U6	LMV339IPWR
U12-U15	BSR14