Controlador de temperatura y calentado para sistemas Skipper-CCD

Federico Gastón Pietra

Director: Javier Tiffenberg
Codirectora: Ana Martina Botti

Tesis de Licenciatura en Ciencias Físicas
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Septiembre de 2022

TEMA:	Controlador de temperatura y calentado para sistemas Skipper-CCD
ALUMNO:	Federico Gastón Pietra
L.U. N°:	677/15
LUGAR DE TRABAJO:	Departamento de Física, FCEN, UBA
DIRECTOR DEL TRABAJO:	Javier Tiffenberg
CODIRECTORA:	Ana Martina Botti
FECHA DE INICIACION:	Septiembre de 2021
FECHA DE FINALIZACION:	Septiembre de 2022
FECHA DE EXAMEN:	Septiembre de 2022

INFORME FINAL APROBADO POR:

Autor	Jurado
Director	Jurado
Profesor(a) de Tesis de Licenciatura	Jurado

Agradecimientos

Quiero agradecer en primer lugar al grupo que conformamos para abordar este proyecto; a Javier y Ana por haber sido mis directores y haberme acompañado durante este año, guiándome y dando sus recomendaciones. Gracias por su paciencia y dedicación! A Miguel y Dario por haber aportado siempre su conocimiento (mención especial a Dario, quien me dió a conocer el proyecto e incentivó a que forme parte a través de mi Tesis). A los grupos de Labo 6 y 7; Miru, Guada, Gonza y Gastón, por haber ayudado a que completemos nuestro objetivo en común.

También quiero agradecer a mi familia, a mi papá y mi mamá, por haberme inculcado desde chico la cultura del estudio y la perseverancia. Siempre atentos a como me iba, apoyándome en el camino que tome.

Gracias a los amigos que me dió la facultad, Agus, Mati y Eze, con quienes compartimos muchas cursadas, horas de estudio, asados y anécdotas.

También agradezco a mis amigos de la vida. Por suerte esta lista sería extensa, pero posiblemente injusta, por si me olvido de alguien. Les agradezco por bancarme y estar siempre pendientes a "¿como viene la carrera?" y "¿cuanto falta para que termines!?!?".

Quiero agradecer especialmente a Anita, quien me incentivó y apoyó en la decisión de empezar esta segunda carrera, aguantando días de estudio y cursadas hasta tarde. Gracias por tu motivación, por enorgullecerte y por 10000 cosas mas!

Resumen

Los sensores CCD (*Charge-Coupled Devices*) son circuitos integrados formados por una red bidimensional de millones de capacitores acoplados llamados píxeles. Estos píxeles son capaces de guardar la carga generada por la interacción de radiación incidente con el material semiconductor y luego moverla secuencialmente hasta un amplificador de salida, donde es leída. Los Skipper-CCD extienden las capacidades de los CCD convencionales, permitiendo realizar múltiples lecturas independientes y no destructivas de la carga de cada pixel y de esa manera bajar el ruido de lectura de 2-5 electrones a una fracción de electrón. Estas características son de particular interés para experimentos de detección de partículas, en especial en aquellos que busquen detectar señales de muy baja energía (del orden de las decenas de eV). Estos dispositivos se operan en vacío y a temperaturas cercanas a 140 K.

En esta tesis desarrollé un sistema de control que permite medir y regular la temperatura del detector, de manera de minimizar los efectos de la corriente oscura que incrementa el error en la detección de partículas. Éste se basa en la implementación de una fuente de potencia regulable de 70 W, cuya tensión y corriente de salida se miden a través de un módulo de telemetría. Además, el sistema cuenta con un mecanismo de disparo de un LED que permite calibrar al sensor y un módulo de control de una electroválvula para regular la salida de vapor de nitrógeno y a su vez la temperatura del sensor.

Como parte del trabajo, se definieron los requerimientos del sistema, y se desarrollaron los algoritmos de control y hardware necesarios para su funcionamiento. También se evaluó su desempeño, instalándolo y tomando datos en uno de los sistemas que ya se encuentran en nuestro país. El resultado demuestra que la utilización del sistema de control no impacta sobre el ruido de lectura, que resultó ser $\sigma = (0{,}69 \pm 0{,}02)e^-$ en toda la escala de potencia, para una medición en modo Skipper con 25 muestras y $\sigma = (0{,}17 \pm 0{,}01)e^-$ para una con 400 muestras.

Índice general

Capítulo 1

Introducción

El presente trabajo se enmarca dentro del programa de colaboración entre el Departamento de Física (DF) de la Universidad de Buenos Aires y el *Fermi National Accelerator Laboratory* (Fermilab) de los Estados Unidos, marco en el cual se creó el Laboratorio Argentino de Mediciones con Bajo umbral de Detección y sus Aplicaciones (LAMBDA), cuyas líneas de investigación versan sobre las ventajas en la utilización de sensores *Skipper-CCD*. En ese sentido, se trabajó en el desarrollo e implementación de un controlador de temperatura de hardware libre que cumpla con los requerimientos necesarios de precisión y bajo ruido de dicho tipo de sistemas. Luego, el objetivo principal de la presente Tesis de Licenciatura consistió en el desarrollo y construcción del sistema de potencia utilizado para controlar la temperatura y el desarrollo en PCB de los circuitos necesarios para su implementación, de manera de poder integrarse con el resto de los módulos que conformarán el sistema de regulación y control de temperatura, en el que, paralelamente, trabajaron dos grupos de laboratoristas 6 y 7.

1.1. Detectores de partículas

Existen diferentes tipos de detectores de partículas, con diversas características y rangos de operación. En general todos tienen en común fundamentalmente dos componentes: un material que auspicia de blanco (*target*) y la electrónica necesaria para medir el resultado de la interacción con una partícula. En esencia, el target es atravesado por partículas que interactúan dentro del mismo depositando parte o la totalidad de su energía, que luego será transformada a una señal eléctrica medible con la electrónica.

Un ejemplo de detector de partículas son los tubos fotomultiplicadores [1], los cuales se utilizan para detectar partículas de luz. Los mismos constan de un fotocátodo (*target*) del cual se liberan electrones por efecto fotoeléctrico luego de la incidencia de un fotón. Los electrones son luego acelerados por un campo eléctrico y multiplicados en una serie de dínodos por efecto avalancha. Ahora bien, este tipo de detectores tienen ciertas limitaciones en cuanto al rango dinámico de operación; por ese motivo y otros, se suelen acoplar ópticamente a otros dispositivos, como son los centelladores, los cuales reciben las partículas a detectar y emiten fotones en el rango de sensibilidad del fotomultiplicador. Estos detectores no cuentan con resolución espacial para los fotones que se quieren medir.

Otro ejemplo de detectores, son aquellos que utilizan semiconductores como material de blanco (típicamente silicio o germanio), donde la zona de depleción o región de vaciamiento de la juntura PN polarizada, es el target de las partículas a detectar [2]. Dichas partículas ionizan el material promoviendo electrones a la banda de conducción, produciendo pares hueco-electrón y generando una corriente libre que luego es posible manipular con la utilización de un campo eléctrico. En la Fig. 1.1 se esquematizan las bandas de energía en un semiconductor, separadas por un *gap*. En (a) se observa la banda de conducción (BC) y un electrón promovido al recibir la energía de un fotón E_γ. De igual forma, se observa la banda de valencia (BV) con el correspondiente hueco del par. En (b), fundamentalmente en el caso del silicio, se puede ver que las bandas no están enfrentadas, por lo que existe un gap indirecto; eso significa que es mucho menos probable que un electrón pase a la BC cuando se recibe un fotón de al menos $E_\gamma = 1{,}12$ eV que cuando el mínimo es 3,4 eV.

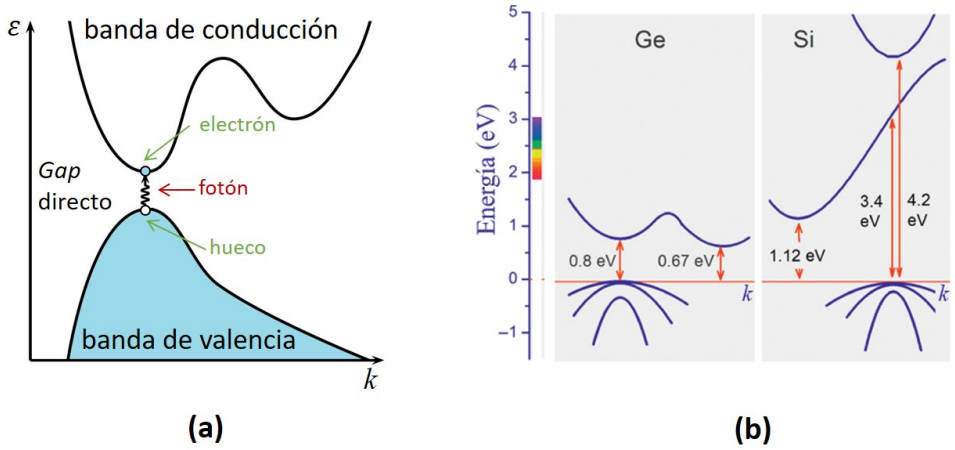

Figura 1.1: Diagrama de bandas de energía en semiconductores: Energía ϵ vs. vector de onda $\bar{k}$. (a) Se observa un par hueco-electrón generado por un fotón que promueve un electrón a la banda de conducción, dejando un hueco en la banda de valencia. (b) Bandas de valencia y conducción y sus valores energéticos para el germanio y el silicio.

Una de las ventajas de este tipo de detectores es que cuentan con una buena resolución en energía, ya que en promedio se necesitan 3,75 eV [3] para generar los mencionados pares hueco-electrón (en silicio). Por otro lado, también es posible acoplar varios de estos detectores formando píxeles, lo cual agrega la posibilidad de tener resolución espacial de las partículas a detectar. Los sensores CCD utilizados en el marco del presente trabajo, pertenecen a la familia de detectores recientemente descriptos.

1.2. CCD y Skipper-CCD

Los CCDs (*Charge Coupled Devices*) nacieron en los Laboratorios Bell en 1969, con el objetivo de ser utilizados como dispositivos de memoria digital, donde la cantidad que representaba un bit de información era la carga almacenada en cada pixel. Si bien esa aplicación no prosperó, demostraron tener potencial como detectores de partículas.

El componente central de estos sensores, son los capacitores MOS (*Metal-Oxide Semicon-*

ductor), de los cuales se compone cada pixel dentro del detector [4]. Luego de un determinado tiempo de exposición, midiendo la tensión en cada pixel y conociendo su capacidad, se puede determinar la carga total por pixel ($V = q/C$). Siendo que típicamente hay millones de estos capacitores, se obtiene de esa forma una gran resolución espacial. De esa forma, los CCDs se componen de una matriz bidimensional de píxeles constituidos por capacitores MOS acoplados, los cuales están compuestos por un sustrato semiconductor dopado, sobre el que se encuentra una capa delgada de óxido, y luego un metal de contacto, donde se aplica una tensión V_G, como se observa esquemáticamente en la Fig. 1.2. Luego, dependiendo del valor de V_G se obtienen distintos regímenes del MOS [5], donde en particular, uno de ellos genera una región de depleción (sin cargas libres) cerca del óxido, el cual permite acumular carga minoritaria.

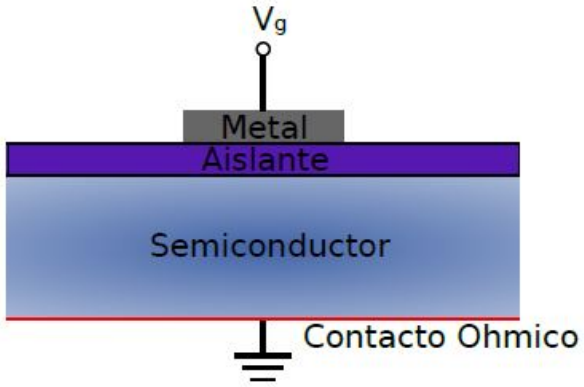

Figura 1.2: Esquema de un capacitor MOS del cual se componen los píxeles de un detector CCD.

El funcionamiento de un CCD comienza con la exposición. En ese proceso, las partículas que atraviesan el sustrato ionizan a los átomos de silicio generando pares electrón-hueco; dichas cargas luego son arrastradas por un campo eléctrico (Fig. 1.3) hasta los pozos de potencial que evitan que se difundan entre píxeles vecinos.

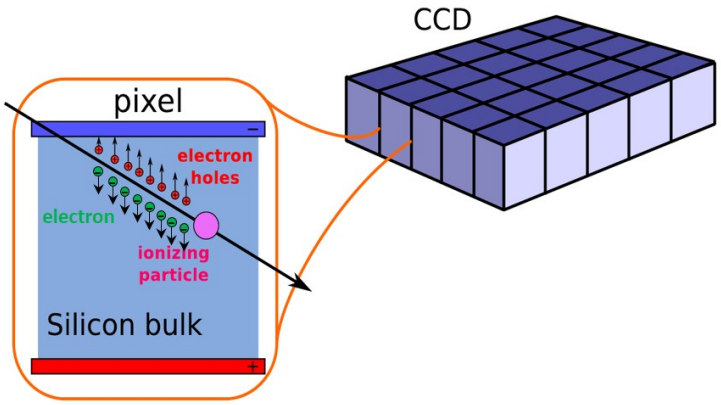

Figura 1.3: Esquema de interacción de una partícula con el semiconductor y colección de cargas generadas por la energía de dicha partícula.

Luego de la exposición y colección de carga, mediante la utilización de pozos de potencial se procede a mover dichas cargas a través de la superficie del CCD para su lectura secuencial, como se esquematiza en la Fig. 1.4. Los *channel-stops* generan barreras de potencial para que la carga no se pueda mover lateralmente a píxeles de columnas vecinas, las señales o relojes V_1, V_2, V_3 (compuertas verticales) generan el desplazamiento de las deposiciones de carga hacia el registro horizontal y los tres canales de desplazamiento horizontal (H_1, H_2 y H_3) mueven la carga dentro del registro serial hacia el nodo de censado de salida, quien consecuentemente tendrá la información de cargas correspondiente a cada pixel.

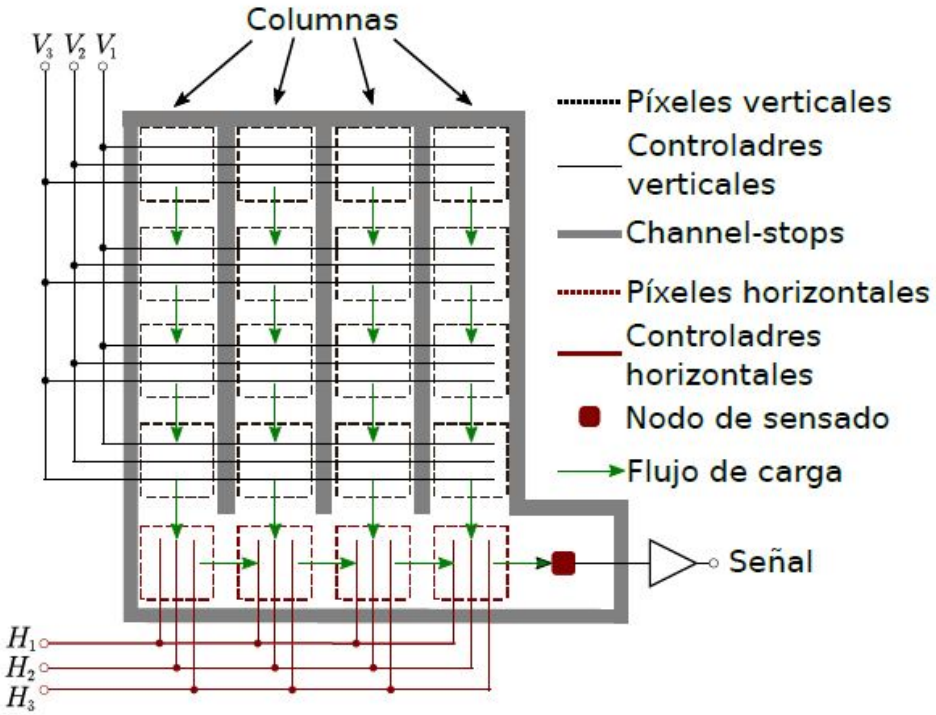

Figura 1.4: Esquema de sensor CCD de 4x4 píxeles, representados en líneas punteadas. Las tres compuertas V_1, V_2, V_3 permiten el desplazamiento de cargas en forma vertical y las H_1, H_2, H_3 los desplazamientos horizontales hacia el nodo de censado de salida. Las flechas verdes muestran la dirección en la que la carga es desplazada pixel a pixel.

En la Fig. 1.5 se muestra esquemáticamente las fases de lectura de un CCD paso a paso. En el paso 1, una partícula ionizante incide sobre el píxel superior derecho generando 3 cargas. En el paso 2, las cargas contenidas en cada píxel son desplazadas hacia la izquierda (una posición). En el paso 3, las cargas son desplazadas hacia abajo en el registro serie y el *sense node* recibe la información del primero de los píxeles. Luego el proceso continúa para leer el resto de los píxeles.

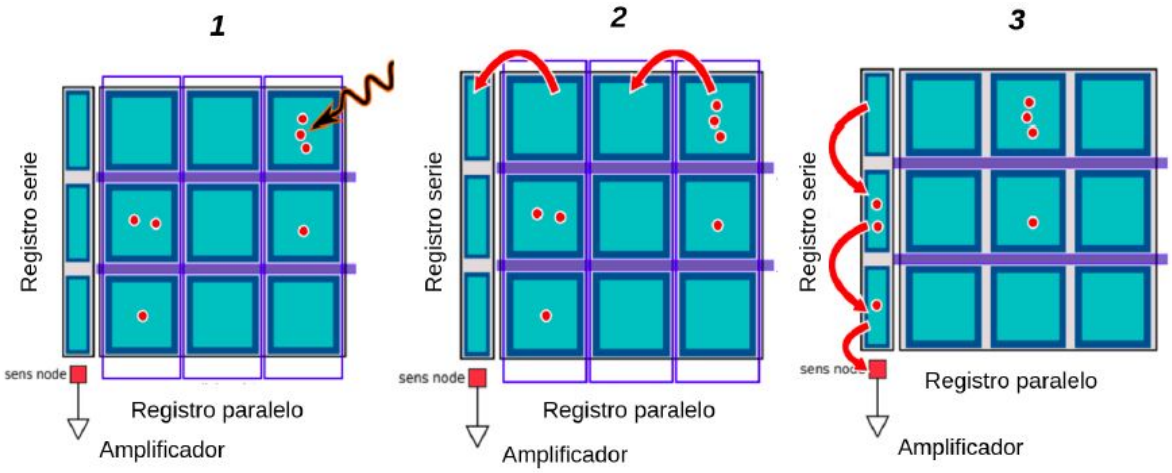

Figura 1.5: Fases de lectura secuencial de un sensor CCD donde se representa el registro serie y el registro paralelo. En el vértice inferior izquierdo se encuentra el amplificador, hacia el cual se dirige la carga de forma secuencial para su lectura. En (1) la interacción ioniza carga, en (2) la carga de toda una fila se mueve a la siguiente y en (3) se observa que una vez en el registro serie, se mueve la carga hasta el amplificador.

Los CCDs de uso científico son capaces de alcanzar ruidos de lectura del orden de los $2e^-$ por pixel [6], debajo del cual se llega a la barrera impuesta por el ruido 1/f (también llamado ruido rosa o *flicker*). Luego, con el objetivo de reducir el ruido aún mas, se desarrolló la técnica de *Skipper-CCD* [7, 8].

En esencia, un Skipper-CCD consiste en un detector CCD con capacidad de realizar múltiples mediciones no destructivas [9] sobre la carga de un mismo pixel. Para ello, se incorpora una compuerta flotante en la zona de lectura del CCD. A través del control de distintos potenciales, se logra que las cargas se desplacen hacia adelante y hacia atrás permitiendo medir reiteradas veces sobre la misma muestra, sin destruirla [7, 10]. De esa forma se puede realizar un promedio de la medición ($\mu = \sum_{i=1}^{N} \frac{q_i}{N}$, con q_i i-ésima medición de la carga de un pixel) y por lo tanto reducir sustancialmente el ruido de lectura. Si un CCD convencional tiene un desvío estándar σ_0, luego un Skipper-CCD tiene $\sigma = \frac{\sigma_0}{\sqrt{N}}$, para N mediciones realizadas (lo cual es válido si los sucesos no son correlacionados). Por ejemplo, para $\sigma_0 = 3{,}5e^-$ y $N = 300$ mediciones, se obtiene $\sigma = 0{,}2e^-$ con un Skipper-CCD.

En la Fig. 1.6 (a) se observa el espectro de carga medida en cada pixel con un Skipper-CCD utilizando N=1 (verde), N=10 (azul), N=200 (rojo), N=1000 (negro) muestras. La mejora en la reolución al aumentar el número de muestras es evidente: en el caso de N=1 es $3{,}5e^-$ y para N=1000 es tan bajo como $0{,}15e^-$. De igual forma, en la Fig. 1.6 (b) se puede ver como el ruido de lectura disminuye con el número de muestras ($\frac{\sigma_0}{\sqrt{N}}$), a medida que el tiempo de lectura aumenta.

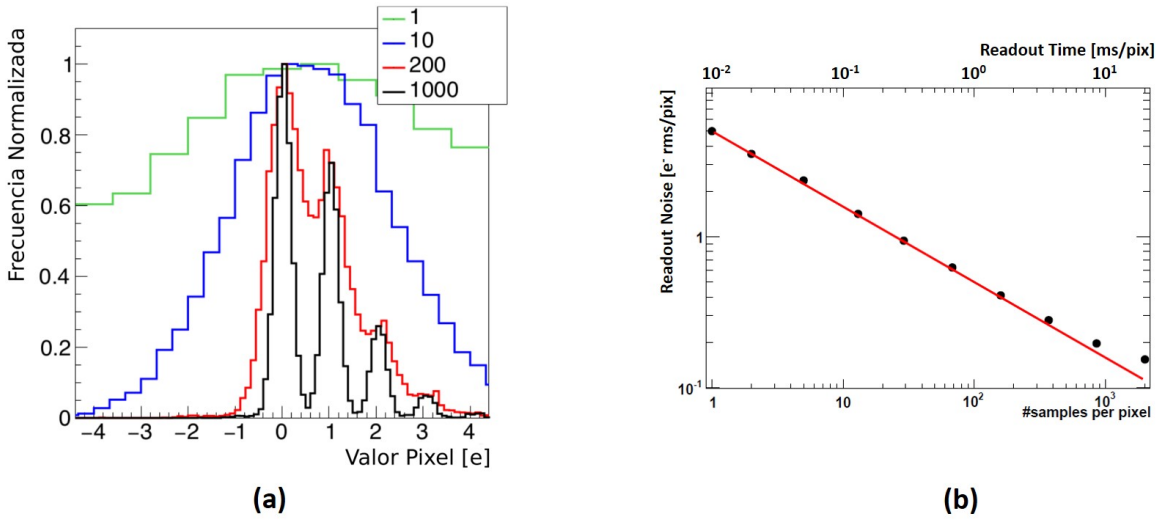

Figura 1.6: (a) Mediciones realizadas con distinto número de muestras promediadas por pixel (1, 10, 200 y 1000). Para 1000 muestras, el ruido de lectura es $0{,}15e^{-}$. (b) Ruido de lectura (desviación estándar) en función del número de muestras por píxel. También se observa que el tiempo de lectura incrementa linealmente con el número de muestras [7].

La reducción del ruido de lectura lograda con los Skipper-CCD permite detectar eventos de bajo umbral energético como los que muestra la Fig. 1.7, donde se observan distinto tipo de partículas como electrones, muones, partículas alfa, etc.

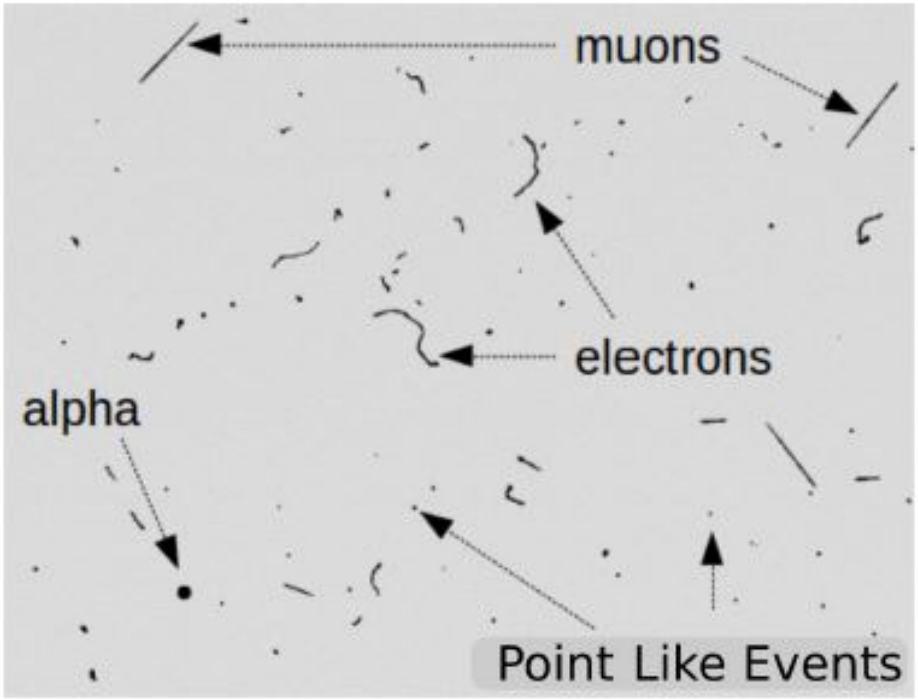

Figura 1.7: Eventos correspondientes a interacciones de distintas partículas detectadas con Skipper-CCD. En la imagen se observan las trazas provenientes de interacciones de muones, electrones, partículas alfa, y otros.

1.3. Aplicación para la detección de neutrinos y materia oscura

Gracias a su bajo umbral de detección, los Skippers son una buena tecnología para la detección de neutrinos o realizar búsqueda de materia oscura (DM - *Dark Matter* - por sus siglas en inglés).

La primera de ellas es una de las partículas elementales que abundan en la naturaleza, no posee carga, tiene una masa muy pequeña, y su probabilidad de interacción con otras partículas es mínima, por lo que se hace muy difícil detectarla. Para cuantificar cuan baja es esa interacción, basta decir que de cada 10^{11} neutrinos que atraviesan el planeta Tierra, solo uno interactúa con la materia bariónica. En ese sentido, en algunos casos es conveniente estudiar su interacción con otra partícula, como el protón (que dió lugar, en 1956, al descubrimiento experimental del antineutrino electrónico: $\bar{\nu_e} + p^+ \rightarrow n^0 + e^+$), o bien su interacción coherente con un núcleo, cuyo proceso esta contemplado en el Modelo Estándar de partículas.

Este último proceso de dispersión, denominado CEνNS por sus siglas en inglés (*Coherent Elastic Neutrino-Nucleus Scattering*), tiene la ventaja de contar con una sección eficaz que escala con el cuadrado del número de neutrones en el núcleo (Fig. 1.8 (b)), y por lo tanto permite detectarla con experimentos de pocos gramos de masa activa, lo que consecuentemente conlleva a que los detectores sean de menor tamaño. En contrapartida, se deposita menor cantidad de energía en la materia, lo cual dificulta su detección. Las interacciones neutrino-electrón tienen la ventaja, por sobre las CEνNS, de no depender del *quenching-factor* (relación entre la energía utilizada para ionizar y la energía transferida al núcleo) del material, el cual repercute en la cantidad de energía transferida en ionización.

Adicionalmente, en la Fig. 1.8 (a) se grafica el número de eventos de las interacciones de neutrinos con distintos elementos dentro de un átomo de silicio, en función de la carga depositada. La curva negra muestra la proporción de eventos neutrino-núcleo ($\nu - N$) predicho por el modelo estándar, mientras que las otras curvas corresponden a posibles nuevas interacciones fuera de la teoría de dicho modelo (se las llama conjuntamente teorías BSM - *Beyond Standard Model*) [11]. Allí se puede observar que las energías depositadas son bajas, por lo que es ventajoso contar con un mecanismo o dispositivo capaz de detectarlas, lo cual da lugar a la utilización de los sensores Skipper-CCD. En particular en ese gráfico, las interacciones $\nu - e^-$ en BSM, son las que mas se beneficiarían de este tipo de tecnologías.

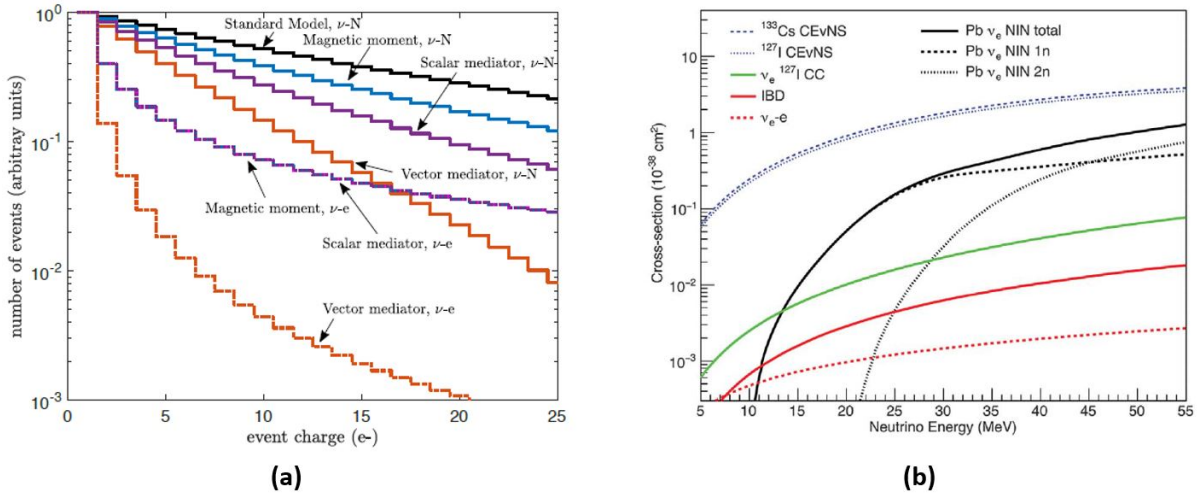

Figura 1.8: (a) Proporción de eventos generados por distintas interacciones en átomos de silicio vs. la carga depositada [3]. (b) Sección eficaz vs. energía del neutrino: en interacciones CEνNS se tiene una mayor sección eficaz, comparado con otras interacciones [12].

El interés por estudiar neutrinos es motivado por el hecho de que si bien se trata de partículas pertenecientes al modelo estándar, de las cuales se conoce cierta información, hay aún mucha física por descubrir, por ejemplo: no se sabe si son su propia antipartícula, no se sabe si hay mas de los tres sabores conocidos (ν_τ, ν_μ y ν_{e^-}), podrían interactuar a través de nuevos mediadores, podrían ser parte de la materia oscura, etc.

En ese sentido, algunos de los experimentos concebidos para la detección de neutrinos de baja energía que utilizan CCDs son: CONNIE (*Coherent Neutrino-Nucleus Interaction Experiment*) [13], quien a través de neutrinos producidos en reactores nucleares, busca entender diferentes interacciones posibles con núcleos de silicio (CEνNS) y otro tipo de interacciones no estándar, o νIOLETA (*Neutrino Interaction Observation with a Low Energy Threshold Array*), que apunta a estudiar nuevas interacciones de neutrinos en teorías BSM, para lo cual se instaló un sensor de tipo Skipper a tan solo 8 metros de la pared del núcleo del rector en la Planta Nuclear Atucha II [14].

En cuanto a la materia oscura, existen diferentes evidencias experimentales que apuntan a su existencia, como ser las curvas de rotación de galaxias [15] y los lentes gravitacionales [16], entre otras. En ese sentido, se estima que el universo esta compuesto por 4,8 % de materia bariónica, 25,8 % de materia oscura no bariónica y 69,2 % de energía oscura [17], motivo que impulsa los esfuerzos por investigarla. Dentro de las teorías que conjeturan su naturaleza, se postulan diferentes masas e interacciones, fuera del modelo estándar [18]. En particular, en aquellos modelos donde la materia oscura interactúa sólo con la materia bariónica a través de la interacción débil, las partículas son conjuntamente denominadas WIMP (*Weakly Interacting Massive Particles* o partículas masivas de inteacción débil) [19, 20]. Entonces, el estudio de este tipo de partículas obedece a una metodología de detección directa, ya que como se mencionó, parte de la base de asumir que interactúan directamente con partículas conocidas del modelo estándar.

Algunos de los experimentos concebidos para la detección directa de materia oscura son: DAMIC (*Dark Matter in CCDs*), el cual utiliza sensores CCDs como blanco para la búsque-

da principalmente de WIMPs con masas < 10 GeV/c^2 [10, 21], o SENSEI (*Sub-Electron-Noise Skipper-CCD Experimental Instrument*), donde ese utiliza un Skipper-CCD instalado en el laboratorio SNOLAB (en Canadá), a 107 metros de profundidad, con el objeto de realizar mediciones de materia oscura liviana con masas menores al GeV [8, 10].

Se estima que estas partículas livianas interactúan con los electrones de los detectores y generan una diferencia de carga medible. Siendo que su masa sería cercana a la del electrón, se espera que las interacciones se manifiesten mediante deposiciones de unas pocas cargas y por lo tanto sea mas probable encontrar eventos de uno o dos electrones [22] (SEE - *Single Electron Event*). Es por ese motivo que se necesita contar con un dispositivo con sensibilidad suficiente para detectar este tipo de eventos (Skipper-CCD), pero también resulta importante determinar, cuantificar y minimizar otros procesos que son capaces de producir eventos similares, como ser los rayos cósmicos, la electrónica utilizada, la temperatura de operación, la corriente oscura, etc.

1.4. Inconvenientes en mediciones de bajo umbral

Dentro de las variables que influyen en los eventos de fondo (*background*) que se quieren minimizar o eliminar, se encuentra la temperatura de operación. Para ello, el rango de temperaturas donde el Skipper-CCD funciona de mejor manera es de 110 a 150 K. Fuera de esa ventana de temperaturas, aparecen efectos indeseados que comprometen las mediciones. Por ejemplo, por debajo de ese rango, la transferencia de carga de un pixel a otro pierde eficiencia. Ahora bien, si nos encontramos por encima de esas temperaturas, predominan los efectos espurios producidos por la denominada corriente oscura (DC - *Dark Current*).

El concepto de DC, de forma general, hace referencia a la señal que se obtiene cuando el detector no esta expuesto a ninguna fuente de luz o partículas, y aparecen de manera espontánea pares electrón-hueco en el silicio. Esos pares pueden generarse por agitación térmica, por la aplicación de campos eléctricos para transportar carga o por defectos en el propio Si del detector.

Entonces, para controlar los efectos producidos por la agitación térmica del silicio, los CCDs son enfriados con un *cryocooler* no regulable que sólo permite disminuir la temperatura a aproximadamente 70-80 K. Luego, para establecer el punto óptimo de operación de los CCDs ($\simeq$140 K), se utiliza una fuente de calor (*heater*) de 40 W que permite controlar la temperatura del sistema y garantizar la estabilidad del mismo; es por esa razón que resulta muy importante contar con un dispositivo capaz de medir y controlar la temperatura de funcionamiento. Si bien actualmente eso se resuelve utilizando un *Lakeshore*, el cual tiene un costo aproximado de USD 7000, la idea y motivación del presente trabajo es diseñar una solución alternativa, de bajo costo y de HW/SW libre de manera de poder resolver la misma problemática.

1.5. Organización de la tesis

En este capítulo se presentó en forma introductoria a los temas que dan contexto y motivación a la realización de esta Tesis; se incluye una breve descripción de algunos detectores de partículas y fundamentalmente se describen los CCD y Skipper-CCD, junto con su aplica-

ción en experimentos desarrollados para la detección de neutrinos y materia oscura. También se mencionan las consideraciones de funcionamiento necesarias para minimizar los eventos de fondo y reducir el ruido de lectura, para lo que es necesario llevar control de la temperatura de operación del Skipper-CCD.

En el capítulo 2 se comienza con una descripción general del sistema Skipper-CCD, para luego adentrarse en la descripción de los distintos módulos y componentes que integran el controlador de temperatura.

En el capítulo 3 se explica el camino seguido en el desarrollo puntualmente de dos de los módulos: la fuente de alimentación conmutada y la telemetría. Además se enumeran las consideraciones y recomendaciones necesarias para las distintas implementaciones probadas, hasta llegar a la versión definitiva utilizada.

El capítulo 4 presenta información referente a la construcción del *hardware*, desde su versión de pruebas en *protoboard*, luego en placa de cobre perforada, para finalmente llegar a la implementación de la versión definitiva en PCB, diseñado en KiCAD. También se describe el *software* de control que se desarrolló en python con el objeto de controlar los distintos periféricos.

En el capítulo 5 se incluye información referente a las mediciones de ruido de lectura realizadas cuando se incorpora el sistema de control desarrollado y se efectúa una detección de partículas, de forma de determinar si su presencia influye o no en dicho resultado.

Capítulo 2

Sistema de control

Este capítulo comienza con una descripción general del sistema que contiene al Skipper-CCD y cuales son sus componentes, a los efectos de plantear el escenario donde se incorpora el dispositivo desarrollado. Luego se describen individualmente cada uno de los módulos que lo conforman, junto con la vinculación existente entre ellos.

Los módulos en cuestión son: el microcontrolador Raspberry Pi Pico, medición de temperatura, algoritmo PID (*Proportional Integral Derivative*), fuente de potencia, telemetría, control LED, control válvula y display. En particular, lo que refiere al desarrollo de la fuente y telemetría se abordará con mayor detalle en el capítulo 3. Vale mencionar que tanto el módulo algoritmo PID como parte del control válvula, fueron desarrollados por los grupos de laboratorio 6 y 7, motivo por el cual no se describirán exhaustivamente.

2.1. Funcionamiento general

En la Fig. 2.1 se observa un esquema del sistema completo que contiene al Skipper-CCD. Éste se encuentra solidario a un bloque de cobre y dentro de una cámara de vacío, junto a una resistencia de platino PT-100 para medir temperatura y una resistencia de 10 Ω para regular la temperatura del sistema (*heater*), un LED para calibración y un circuito de enfriado a base de nitrógeno líquido. Se lleva al sistema a una temperatura aproximada de 80 K, a través de la regulación manual de una válvula de control. Luego, a través del heater mediante un algoritmo PID se calienta el sistema a aproximadamente 120-140 K. Para llevar a cabo dicho procedimiento es necesario evitar que la temperatura varíe a una tasa mayor a 1 K/min, ya que eso podría dañar al sensor.

El sistema cuenta con una bomba de vacío para evitar condensación de la humedad presente en el aire sobre los detectores, lo cual podría formar una capa de hielo degradando o dañando el detector. Finalmente, la electrónica de lectura consiste en una LTA (*Low Threshold Adquisition*)[23]; un sistema de adquisición de imágenes desarrollado para el experimento SENSEI [8].

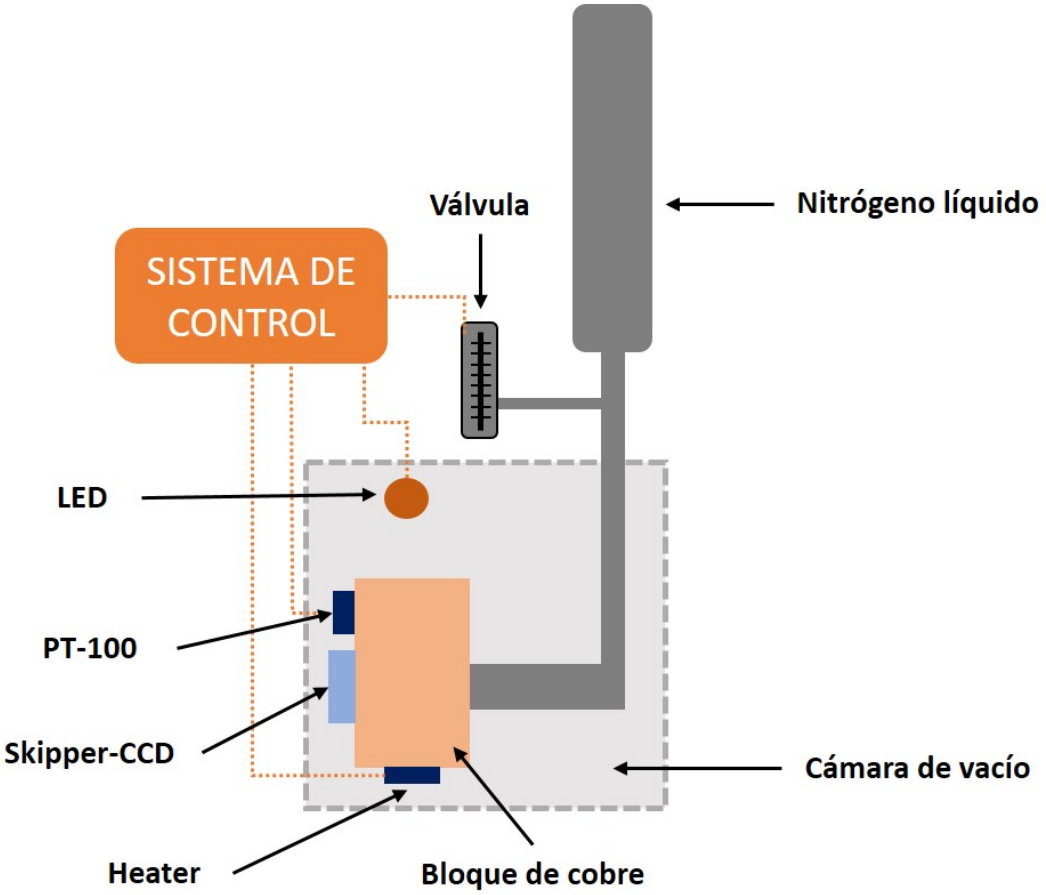

Figura 2.1: Diagrama esquemático del sistema que contiene al Skipper-CCD. Se representa el sensor CCD propiamente dicho, el *heater* y el sensor de temperatura PT-100 ubicados dentro de la cámara de vacío. También se esquematiza el contenedor de nitrógeno líquido y su válvula de regulación de vapor de nitrógeno y finalmente el sistema de control desarrollado.

En la Fig. 2.1 se observa también el **sistema de control** desarrollado en este trabajo de tesis, el cual fundamentalmente controla la temperatura del sistema. Sensamos dicha temperatura con una resistencia PT-100, y se regula el heater para mantenerla entre 120 y 150 K, que resulta ser rango de desempeño óptimo del Skipper-CCD. Además, el controlador incluye un mecanismo que permite abrir/cerrar la válvula de forma remota, evitando el accionamiento manual, y un algoritmo que permite controlar la intensidad y tiempo de encendido de un LED utilizado para calibrar la CCD.

En la Fig. 2.2 se exhibe una foto del sistema completo instalado y funcionando que se encuentra en el Laboratorio LAMBDA. Se observa la bomba de vacío, el termo de nitrógeno líquido junto con la válvula mecánica de control de vapor de nitrógeno y la cámara de vacío que contiene al Skipper-CCD.

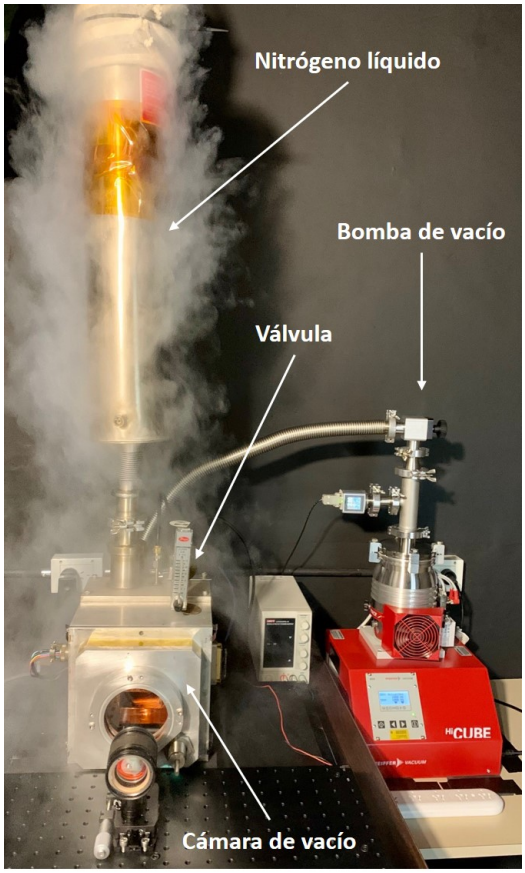

Figura 2.2: Sistema completo que contiene al Skipper-CCD, instalado en el Laboratorio LAMBDA.

En la Fig. 2.3 se esquematiza específicamente el sistema de control antes descripto, con los diferentes módulos que conforman la solución completa. En el módulo **medición de temperatura** implementamos una resistencia de platino PT-100 instalada en la cámara de vacío. Luego el **PID** utiliza esa información para regular la fuente, definiendo la potencia a la cual debe operar el *heater*, y converger mas rápidamente a la temperatura deseada. Además, el módulo de **telemetría** mide la tensión y corriente de dicha fuente, con el objeto de que el sistema cuente con un método para validar su correcto funcionamiento. Por otro lado, se observa el **control LED** y **control válvula**, cuyas funciones se mencionaron anteriormente. Finalmente, el sistema cuenta con un **display** OLED, el cual permite mostrar información relacionada a los otros módulos (temperatura, potencia de la fuente, intensidad del LED, etc).

En las subsecciones a continuación, se detallan los componentes y módulos principales del sistema.

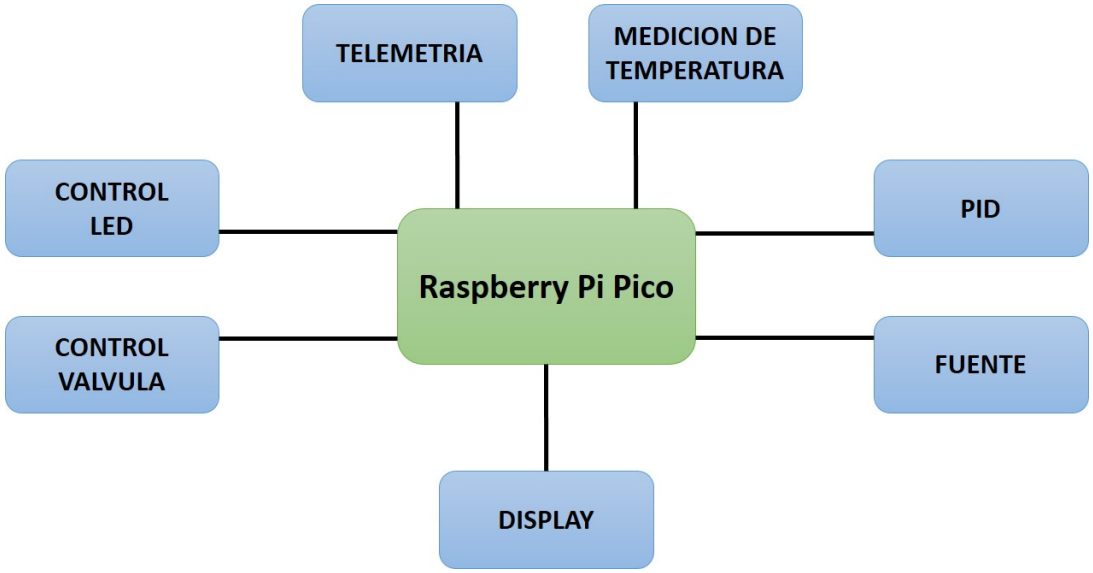

Figura 2.3: Diagrama en bloques del controlador de temperatura desarrollado. En verde la Raspberry Pi Pico, el componente que comanda y controla al resto de los módulos (en azul).

2.2. Microcontrolador Raspberry Pi Pico

El componente central en el diseño del sistema de control se basa en la utilización del microcontrolador Raspberry Pi Pico [24]. Esta placa de desarrollo esta basada en el chip RP2040 diseñado por Raspberry Pi Foundation, el cual cuenta con dos cores ARM Cortex-M0+ que trabajan a 133 MHz, memoria Flash de 2 MB y memoria RAM de 264 KB. La Raspberry Pi Pico tiene integrada una fuente de alimentación conmutada (SMPS - *Switch Mode Power Supply*) buck-boost que permite generar lo 3,3 V requeridos para alimentar al RP2040 a partir de un rango de voltajes de entrada (de 1,8 a 5,5 V). Además cuenta con 30 pines de entrada y salida de propósito general (GPIO - *General Purpose Input/Output*), de los cuales 26 están disponibles y los 4 restantes se utilizan para funciones internas: para controlar un led integrado en la placa, para la fuente de alimentación conmutada y para detectar los voltajes del sistema. Los pines de entrada y salida en la Raspberry Pi Pico trabajan a 3,3 V y pueden utilizarse de distinta manera; 2 SPI (*Serial Peripheral Interface*), 3 ADC (*Analog-to-Digital Converter*, 2 I2C (*Inter-Integrated Circuit*), 2 UART (*Universal Asynchronous Receiver-Transmitter*), 16 canales PWM (*Pulse width modulation*). En la Fig. 2.4 se observa el pinout de la Raspberry Pi Pico, con la posición de los pines descriptos.

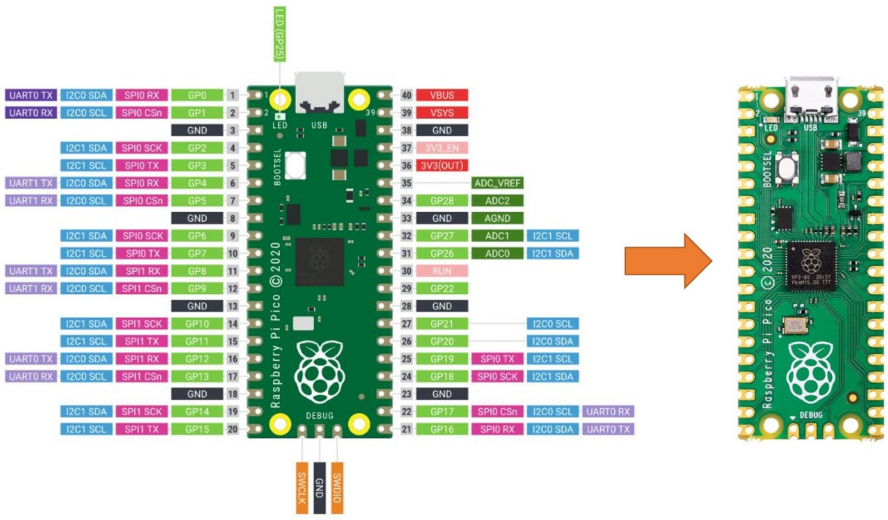

Figura 2.4: Izquierda: Esquema de microcontrolador Raspberry Pi Pico y sus diferentes tipos de puertos. Derecha: Foto del microcontrolador.

Éste microcontrolador se conecta a una PC con su puerto de conexión USB y se puede programar en C/C++ o MicroPhython: un compilador de Phython orientado al uso en microcontroladores. En este caso, se optó por la segunda opción. En particular, el entorno de desarrollo integrado (IDE - *Integrated Development Environment*) elegido fue Thonny, donde fácilmente se puede instalar el intérprete deseado (MicroPhython) en la Raspberry, al igual que las librerías necesarias, y escribir los códigos para su funcionamiento. En la Fig. 2.5 se muestra una captura de pantalla de Thonny; a la izquierda figura el explorador local donde se encuentran los archivos y abajo el explorador dentro de la Raspberry, en la parte central el código que se esta escribiendo y abajo el resultado luego de su ejecución, y finalmente a la derecha un listado de variables y sus valores.

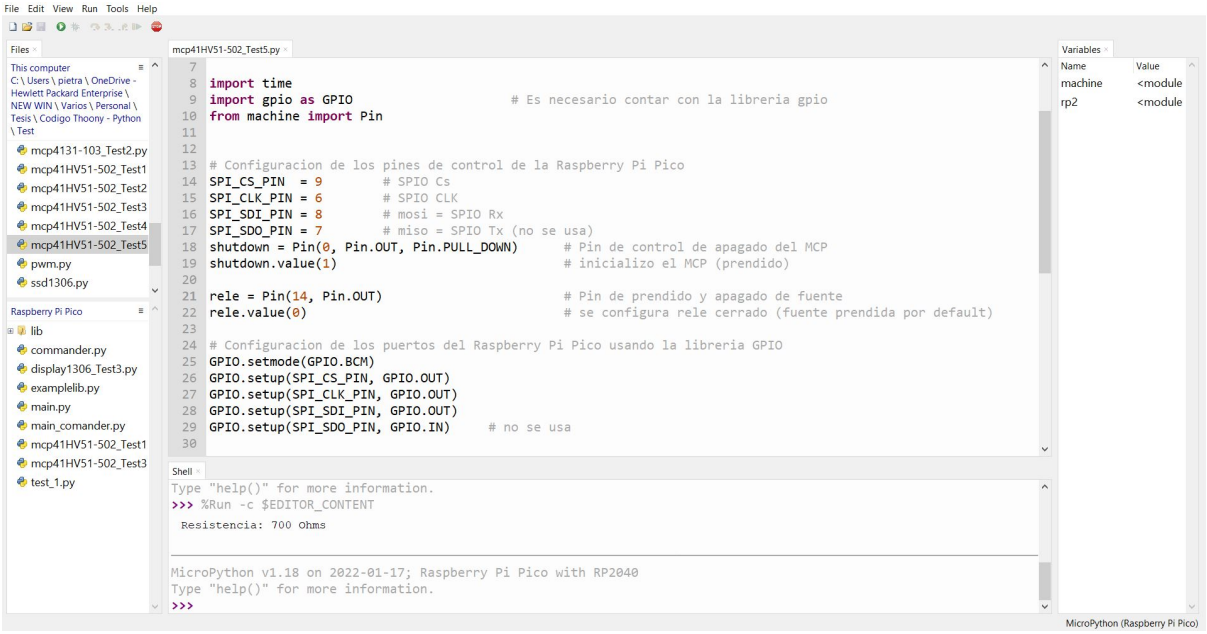

Figura 2.5: Captura de pantalla de Thonny: Entorno de desarrollo integrado (IDE) utilizado en el proyecto.

2.3. Medición de temperatura

Con el objeto de medir la temperatura del sistema con precisión se utiliza un sensor RTD (*Resistance Temperature Detector*), en particular una resistencia de platino PT-100 cuya dependencia con la temperatura esta universalmente caracterizada y es 100 Ω a 0 °C. Para leer y convertir el valor de resistencia de la PT-100 a temperatura, usamos un conversor analógico digital Max31865 [25], el cual se muestra en la Fig. 2.6. Su función es la de convertir y amplificar el valor de la resistencia, conectándose con la Raspberry a través del protocolo SPI, el cual esencialmente es un protocolo de comunicación serial sincrónico que trabaja en modo full duplex, permitiendo que dos dispositivos puedan comunicarse entre sí simultáneamente.

Para ello se define un *master* (en este caso la Raspberry) y un *slave* (la Max31865), y de esa forma se conectan físicamente los pines de un dispositivo con los del otro, como se ejemplifica en la Fig. 2.7. Entonces, el master envía datos a través de MOSI (*Master Out Slave In*) mientras que el slave responde a través de la línea MISO (*Master In Slave Out*), sincronizado a través del reloj SCK (*Serial Clock*). Luego, con el pin SS (*Slave Select*) se selecciona con cual slave quiere comunicarse: SPI permite la comunicación de un master con varios slave. En este caso sólo utilizamos uno.

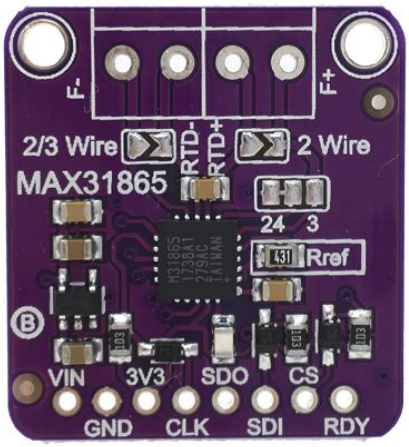

Figura 2.6: Conversor analógico-digital Adafruit Max31865 utilizado como interfaz entre un sensor PT-100 y el microcontrolador Raspberry Pi Pico.

Raspberry Pi Pico		Max31865	
MOSI *Master Out Slave In*	Pin 25 - GPIO_19	SDI *Serial Data Input*	Pin 6
MISO *Master In Slave Out*	Pin 21 - GPIO_16	SDO *Serial Data Output*	Pin 5
SPIO SCLK *Clock*	Pin 24 - GPIO_18	CLK *Clock*	Pin 4
3V3 *Power supply*	Pin 36	VIN *Power supply*	Pin 1
GND *Ground*	Pin 38	GND *Ground*	Pin 2
SPIO CSn *Chip select*	Pin 22 - GPIO_17	SS *Slave select*	Pin 7

Figura 2.7: Diagrama de conexionado entre Raspberry Pi Pico y Max31865.

A los efectos de medir la temperatura para luego regularla, se escribió un código (se puede encontrar en el repositorio `https://gitlab.com/koala13/tesis`) que permite leer la PT-100 y calcular la temperatura, para luego imprimirla en pantalla junto con la fecha y horario en que se tomó cada medición. Adicionalmente se incorporó la funcionalidad de que el control de dichas mediciones se pueda comandar en forma remota a través de un puerto simil serial, utilizando la librería de Python `pyserial`. De esa forma, teniendo conectada la Raspberry al puerto USB de una PC local, se pueden enviar comandos sin la necesidad de tener instalado y utilizar por ejemplo Thonny. Se implementó de tal manera de que el usuario pueda elegir el intervalo de tiempo entre cada medición, por ejemplo: `#python test.py -t -d 10`, mide la temperatura cada un DELTA "d" de 10 segundos, y guarda los registros de fecha/horario, temperatura y valor de RTD, en un archivo *temperaturas.csv*.

Antes de la implementación de este nuevo sistema de medición, se medía la resistencia de la PT-100 con un multímetro y se convertía a través de una tabla de conversión a °C, luego a K. Ese mecanismo por un lado es impreciso y por otro implica un potencial riesgo sobre el detector, ya que como se indicó, hay que controlar que su temperatura no cambie mas de 1 grado K por minuto. En este trabajo se utilizó el sistema de control desarrollado para medir inicialmente la temperatura de enfriamiento y calentamiento del Skipper-CCD instalado en LAMBDA. En particular y a fines didácticos, se realizaron las mediciones a 2 puntas, aunque bien podría hacerse a 4 puntas, ya que la Max31865 asi lo permite.

En la Fig. 2.8 (a) se expone un esquema de medición a 2 puntas, cuyo funcionamiento se basa en enviar una corriente (I) y medir una tensión (V), luego por Ley de Ohm se calcula la resistencia ($R = VI$). Este método tiene la desventaja a agregar error a la medición, debido a que la resistencia total resulta de la combinación de la resistencia que se pretende medir (R) y todas las resistencias de los cables y las conexiones (R_W). En contrapartida, en la Fig. 2.8 (b) se puede observar una medición a 4 puntas, donde a partir de separar los cables que generan la corriente de los que miden la tensión, se eliminan las contribuciones de las resistencias de cableado y los potenciales de contacto sobre la medición final de la resistencia (R) en cuestión, mejorando de esa forma la precisión.

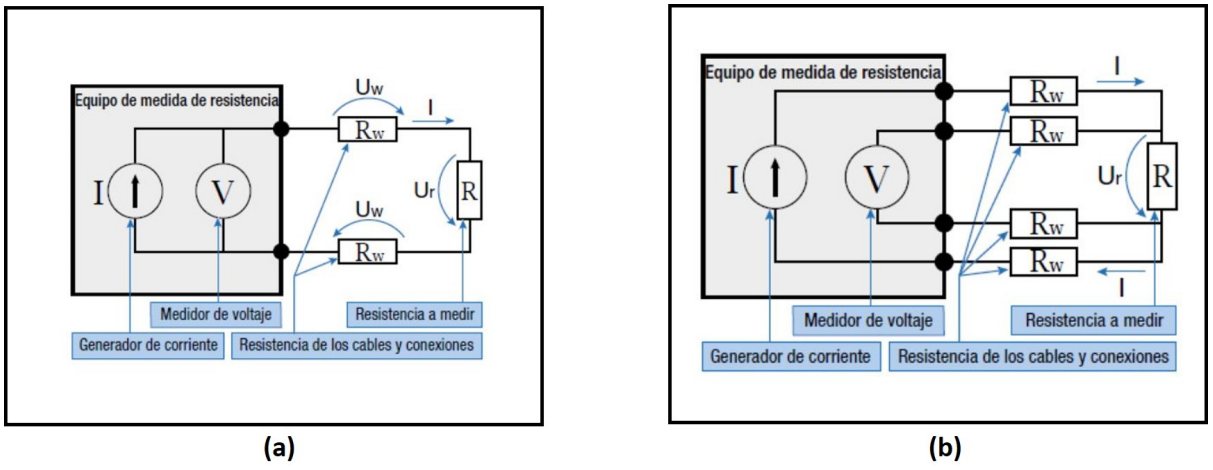

Figura 2.8: (a) Esquema de medición de resistencia R a 2 puntas. (b) Medición de resistencia R a 4 puntas, mejorando la precisión al eliminar la injerencia de la resistencia propia de los cables y conexiones.

Por otro lado, la Fig. 2.9 se muestra un ejemplo de la curva de enfriamiento del Skipper obtenida con el dispositivo y código de control construido. Se observa un extracto de los datos medidos, donde la temperatura baja desde aproximadamente 250 K hasta 175 K en 5 horas, y tiene un comportamiento de tipo exponencial, según ajuste hecho en python.

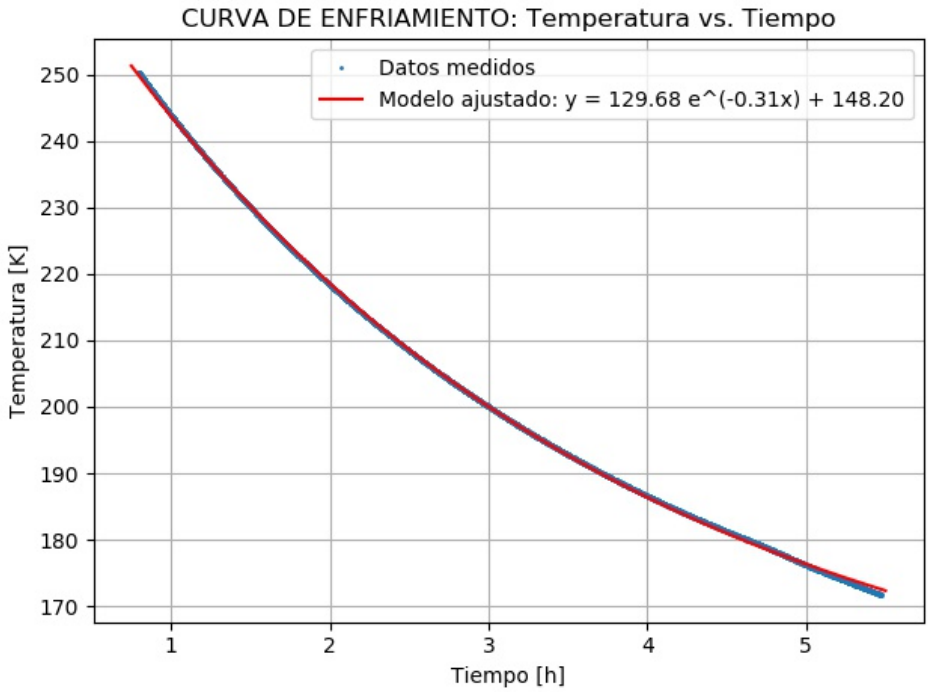

Figura 2.9: Curva de enfriamiento del Skipper-CCD con datos obtenidos con Max31865, ajustados por una curva exponencial decreciente desde 250 K hasta 175 K en aprox. 5 horas.

Adicionalmente, se expone la Fig. 2.10, donde se realizó una captura de pantalla de cómo se exhiben en tiempo real algunas de las mediciones (con fecha/horario y temperatura), correspondientes a la Fig. 2.9.

```
(base) rodriguesfm@minilambda:~/Desktop/Fede_files$ python3 test10.py -t -d 1
Conectado al puerto serial /dev/ttyACM0...

2022-07-28 09:04:46 AM ---> Temperatura: 291.07809 K.
2022-07-28 09:04:47 AM ---> Temperatura: 291.07809 K.
2022-07-28 09:04:48 AM ---> Temperatura: 291.07809 K.
2022-07-28 09:04:49 AM ---> Temperatura: 291.11189 K.
2022-07-28 09:04:50 AM ---> Temperatura: 291.14544 K.
2022-07-28 09:04:51 AM ---> Temperatura: 291.11189 K.
2022-07-28 09:04:52 AM ---> Temperatura: 291.11189 K.
2022-07-28 09:04:54 AM ---> Temperatura: 291.11189 K.
2022-07-28 09:04:55 AM ---> Temperatura: 291.07809 K.
2022-07-28 09:04:56 AM ---> Temperatura: 291.07809 K.
2022-07-28 09:04:57 AM ---> Temperatura: 291.11189 K.
2022-07-28 09:04:58 AM ---> Temperatura: 291.07809 K.
2022-07-28 09:04:59 AM ---> Temperatura: 291.11189 K.
2022-07-28 09:05:00 AM ---> Temperatura: 291.07809 K.
2022-07-28 09:05:02 AM ---> Temperatura: 291.07809 K.
2022-07-28 09:05:03 AM ---> Temperatura: 291.11189 K.
2022-07-28 09:05:04 AM ---> Temperatura: 291.07809 K.
2022-07-28 09:05:05 AM ---> Temperatura: 291.07809 K.
2022-07-28 09:05:06 AM ---> Temperatura: 291.07809 K.
2022-07-28 09:05:07 AM ---> Temperatura: 291.07809 K.
2022-07-28 09:05:08 AM ---> Temperatura: 291.07809 K.
```

Figura 2.10: Datos de fecha/horario y temperatura observados en pantalla como resultado del código de control realizado para la Max31865.

2.4. PID

Un controlador PID [26, 27] es un dispositivo que permite regular un sistema a través de un lazo de realimentación, para que alcance el estado de salida deseado. Como su nombre lo indica, está compuesto de tres elementos que proporcionan una acción **Proporcional**, **Integral** y **Derivativa**, reguladas por sus respectivas constantes K_p, K_i y K_d. Su funcionamiento se basa el cálculo de la diferencia entre un valor medido $h_{(t)}$ y un valor deseado o de referencia $r_{(t)}$, denominado error $e_{(t)}$, a los efectos de minimizarlo y conseguir la señal de salida $y_{(t)}$ buscada. En la Fig. 2.11 se esquematiza un sistema de control basado en PID.

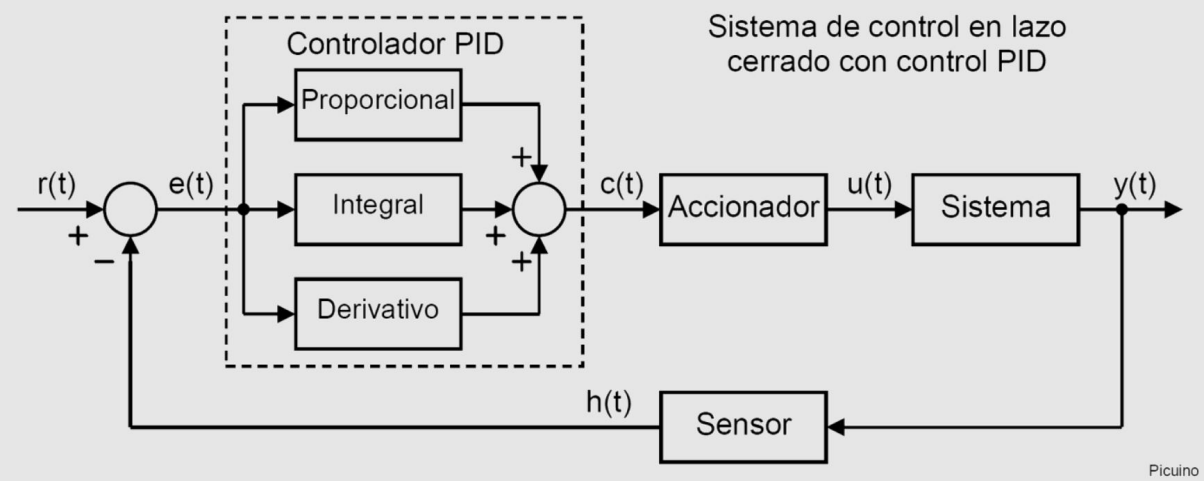

Figura 2.11: Diagrama en bloques de un sistema de control con PID.

Si por ejemplo se utiliza este mecanismo para controlar la temperatura de un sistema [28] (como es nuestro caso), entonces la referencia $r_{(t)}$ será la temperatura deseada y la salida $y_{(t)}$ será la temperatura real del sistema controlado. Luego, si la señal de error $e_{(t)}$ es grande, significa que el estado del sistema se encuentra lejos de la temperatura deseada. Si por el contrario el error es pequeño, significa que el sistema ha alcanzado el valor deseado.

La ventaja principal de la implementación de un PID es que permite converger a la temperatura seteada y hacerlo de forma más rápida a partir de la elección adecuada de las constantes, como se observa en la Fig. 2.12 (b), donde la curva azul indica la evolución de la temperatura hasta llegar al valor deseado (*TEMP. set point*). En contrapartida, en la Fig. 2.12 (a) se representa un sistema que controla la temperatura a través de un mecanismo ON/OFF, entonces cuando la temperatura medida esta por debajo del set point, se prende el calentador. Luego cuando ese punto es superado, se apaga y así sucesivamente, con la desventaja frente a la opción con PID de que nunca converge al valor deseado.

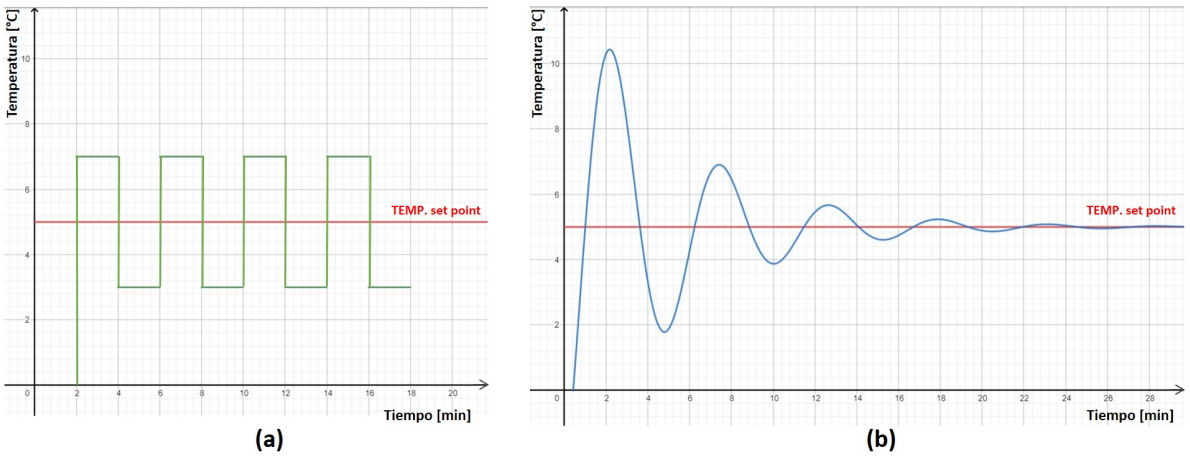

Figura 2.12: (a) Siestema de control de temperatura tipo ON/OFF. (b) Sistema de control de temperatura con PID.

La ecuación de evolución de un sistema PID es la expuesta en la Ec. 2.1, y dependiendo los valores que se adopte para las constantes K_p, K_i y K_d, será el tipo de respuesta que se obtenga. Vale destacar que esas constantes dependerán del sistema específico que se quiera controlar, es decir, de su geometría, material de construcción, ubicación del sensor, condiciones de entorno, etc. Luego, para definirlas existen algunos métodos generales.

$$c_{(t)} = K_p e_{(t)} + K_i \int e_{(t)} dt + K_d \frac{\partial e_{(t)}}{\partial t} \tag{2.1}$$

En la Fig. 2.13 se muestran algunos ejemplos cualitativos de posibles respuestas de un sistema, frente a diferentes valores de sus constantes: en (a) no hay oscilaciones, en (b) las constantes no fueron correctamente elegidas y por lo tengo el sistema nunca llega al set point, en (c) hay pocas oscilaciones y su amplitudes y tiempo de convergencia son mayores a lo obtenido en (d), donde se representa una situación mas favorable.

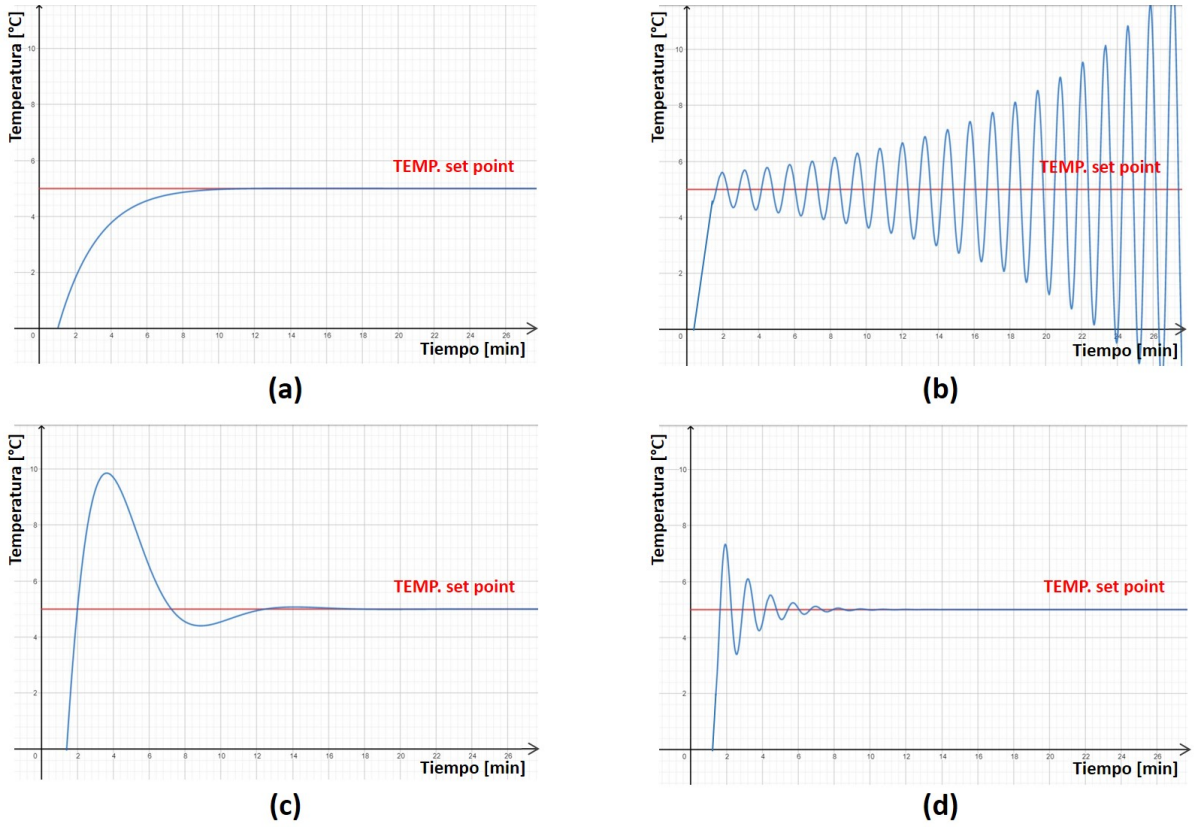

Figura 2.13: Comportamiento de un sistema de control de temperatura PID con distinto valor de constantes K_p, K_i y K_d. (a) Respuesta sin oscilaciones. (b) Sistema que no converge. (c) Pocas oscilaciones de mayor amplitud y tiempo de convergencia. (d) Tiempo de convergencia y amplitud de oscilaciones menores.

Finalmente, la función del PID en nuestro sistema es la de definir la potencia que debe recibir el heater para que el Skipper llegue al *set point* de temperatura que se establezca, y se mantenga en ese rango de forma estable. Mas concretamente, recibe como entrada la temperatura medida y la deseada, y en función de eso comanda la tensión de salida de la fuente. En nuestro sistema de control, el algoritmo PID fue implementado y optimizado por un grupo de laboratorio 6 y 7.

2.5. Fuente de potencia

Este módulo es responsable de entregar la potencia necesaria al heater, de forma tal de regular la temperatura del sistema. Para ello se utilizó un módulo reductor de tensión, como muestra la Fig. 2.14, cuyo componente principal es un circuito integrado XL4016 [29] (*DC to DC Buck Converter*), el cual se eligió por soportar y tener las tensiones y corrientes de salida necesarias (40 V, 8 A) y una alta frecuencia de conmutación (180 KHz), lo cual evita incorporar ruido eléctrico al sistema (indeseado para la detección de partículas). En el capítulo 3 se describirá detalladamente este módulo y las modificaciones realizadas. Aquí sólo se mencionan sus funciones y características generales.

Figura 2.14: Fuente conmutada reductora de tensión (*DC to DC buck converter*) basada en XL4016, utilizada para controlar la potencia del heater.

Esta placa se alimenta con una fuente de notebook comercial, cuya tensión/corriente de salida es 19,50 V/3,42 A (máximo). Consecuentemente, la máxima potencia teórica entregable al heater es P=V.I=(19,50 V)(3,42 A)=66,69 W. Además, cuenta con un potenciómetro que permite regular la tensión de salida de forma manual, motivo por el cual éste fue reemplazado por un mecanismo de regulación remoto basado en un potenciómetro digital de alta tensión MCP41HV51-502 [30] (5 KΩ, 256 pasos). También se incorporó un sistema de encendido y apagado a través de un relé; éste circuito permite fundamentalmente entregar 0 V al heater, ya que la fuente en sí tiene una valor de tensión mínimo residual de $\simeq$ 1,84 V.

El mecanismo descripto, si bien permite configurar el voltage de salida al valor deseado, lo que se hizo fue integrarlo con el PID a los efectos de que el heater reciba la potencia mas conveniente para que la temperatura converja mas rápidamente al valor seteado.

Este módulo se integró con el de telemetría (Sec. 2.6) y con el display (Sec. 2.9), para que el usuario pueda conocer la potencia entregada.

2.6. Telemetría

La función principal de este módulo es la de proveer un mecanismo para informar la potencia que entrega la fuente al heater. Para tal fin, se utilizaron fundamentalmente dos puertos ADC de la Raspberry Pi Pico: uno para medir tensión (ADC_1 - Pin 27) y el otro para la corriente (ADC_0 - Pin 26). Luego de digitalizar ambas señales, se multiplican sus valores (P=V.I) y se informa en el display OLED (descripto en la Sec. 2.9).

En el caso específico de la corriente, fue necesario incorporar un sensor ACS712-5A [31] (Fig. 2.15), cuya función es convertir la corriente que circula por él a una tensión analógica proporcional, que luego es digitalizada con el ADC del microcontrolador.

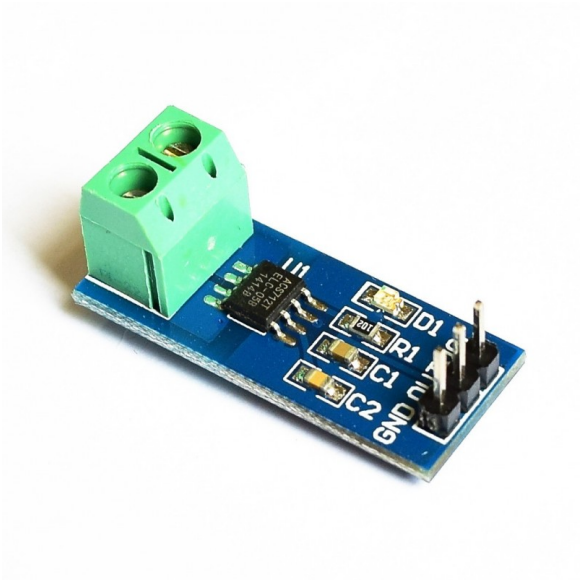

Figura 2.15: Sensor de corriente ACS712 de 5A, basado en efecto Hall, utilizado para la medición de amperaje entregado por la fuente.

Dado que los conversores ADC de estos microcontroladores no son del todo precisos y a su vez tienen algunos problemas, se decidió incorporar un mecanismo de calibración que permite de algún modo cotejar y ajustar los valores medidos con un multímetro vs. los obtenidos con los ADC.

El mecanismo de digitalización e implementación del circuito, junto con el procedimiento de calibración, se explicaran con detalle en el capítulo 3.

2.7. Control LED

El control de LED se incorporó al sistema de medición y regulación de temperatura ya que resulta de utilidad contar un mecanismo de emisión de luz controlada para calibración.

La cámara de vacío donde se encuentra el Skipper-CCD contiene un LED que luego se conecta a nuestra placa (pines: ánodo y cátodo), y a través de una rutina de control en la Raspberry, logramos configurar la intensidad de dicho LED y el tiempo de prendido. Esos parámetros determinan la cantidad de fotones emitidos y, a su vez, la cantidad de electrones ionizados en el Skipper.

El código desarrollado utiliza el PWM (*Pulse-Width Modulation*) de la Raspberry, que permite simular una señal analógica a partir de una digital. Es decir, en el caso de un LED, en lugar de solo prenderlo o apagarlo, se puede regular su intensidad a partir del ciclo de trabajo (*duty cycle*) del PWM. A continuación se transcribe la función `setIntensity` escrita para tal fin, la cual recibe como argumentos a la `intensidad` deseada (de 0 a 100 %) y la `duracion` (cantidad de tiempo prendido, en segundos):

```
0 from machine import Pin                    # clase Pin de libreria machine
1 import time                                # libreria time
2
```

```
def setIntensity(intensidad,duracion):
    pwm = PWM (Pin(15))                          # SPIO PWM para LED, pin 15
    pwm.freq(1000)                               # frecuencia del PWM
    pwm.duty_u16(round(65025*intensidad/100))    # duty cycle del PWM (0
    ↪ apagado, 65025 max intensidad)
    print ('Intensidad del Led:', intensidad,'%') # imprime intensidad en %
    print ('Prendido:', duracion, 'seg\n')       # imprime tiempo ON en [seg]
    time.sleep(duracion)                         # aplica tiempo ON
    pwm.duty_u16(0)                              # apaga el LED
```

En dicha función se crea una instancia de la clase `PWM` con el número de pin a controlar (15), y se definen la frecuencia (`freq`) y ciclo de trabajo (`duty_u16`) de tal objeto. Luego, al igual que para el caso de medición de temperatura con Max31865, aquí también se incorporó la posibilidad de que el control del LED sea por línea de comando a través de una conexión por puerto serial y no necesariamente desde Thonny. En ese sentido, por ejemplo con el comando `#python test.py -l -p 1.3 -i 67.8` , se prende el led un PERIODO "p" de 1,3 segundos, con una INTENSIDAD "i" de 67,8 %. Luego, esa información tambien es informada al usuario en el display, como se verá posteriormente.

2.8. Control válvula

El módulo de control de válvula tiene la función de controlar el sistema de liberación de vapor de nitrógeno. De esta forma, se regula la proporción de nitrógeno líquido en el tubo, variando su contacto térmico con la superficie del cobre al cual se encuentra adosado el Skipper y, de esa forma, se controla el ritmo con el que se enfría el sistema. Anteriormente, ese procedimiento se realizaba en forma manual, accionado una canilla. Entonces lo que se hizo fué incorporar una electroválvula que automatiza el proceso y cuyo control esta comandado por la Raspberry y un circuito basado en un relé, desarrollado para este fin, como muestra la Fig. 2.16. Luego, la descripción de su funcionamiento se hará en el capítulo 3.

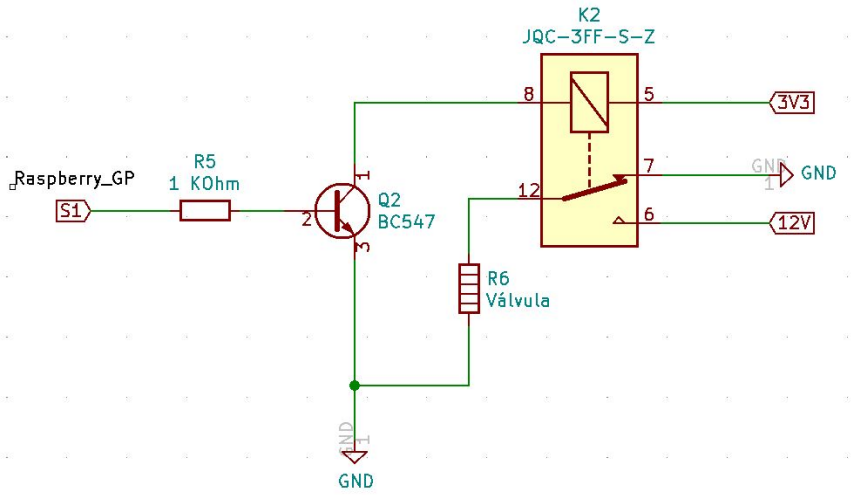

Figura 2.16: Circuito de control de electroválvula encargada de liberar vapor de nitrógeno líquido y de esa forma controlar la velocidad de enfriado del sistema.

La implementación de las funciones mas avanzadas del software de control de este circuito, la realizó uno de los grupos de laboratoristas, razón por la cual no entraré en mayor detalle.

Por otro lado, vale la pena mencionar que el control de nitrógeno aplicado es también un método de regulación de temperatura del sistema. Es decir, o bien se puede regular la temperatura con el heater (y su fuente regulada) o bien permitiendo o no la liberación de vapor de nitrógeno. De esa forma, el control por heater tiene la ventaja de permitir converger mas rápido a la temperatura deseada, ya que se puede suministrar mayor o menor potencia al heater, logrando que caliente o no mas rápidamente. En cambio, al controlar la válvula con relé, solo es posible definir cuando se libera vapor o no, pero al aplicar por demás, hay que esperar un tiempo mayor a que la temperatura llegue al valor deseado (por inercia térmica). En conclusión, es mas inmediato y estable hacer el control con heater.

2.9. Display

Para poder visualizar localmente la información de relevancia obtenida con el sistema de control, se agregó un display OLED SSD1306 de 128x64 píxeles [32] (Fig. 2.17).

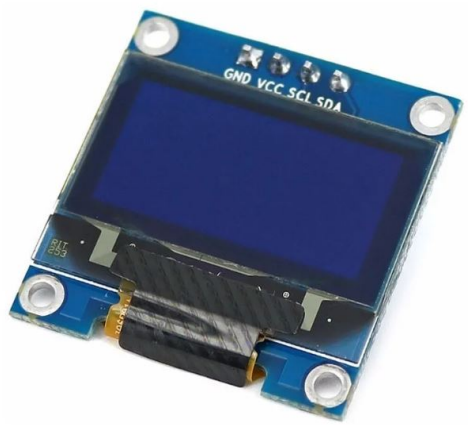

Figura 2.17: Display OLED SSD1306 de 128x64 píxeles.

El acceso al display se realiza usando el protocolo I2C, cuyos pines de acceso son SCL (*Serial CLock*) y SDA (*Serial DAta*). Este protocolo puede admitir múltiples dispositivos slave, pero a diferencia de SPI previamente descripto, que solo admite un dispositivo master, I2C también puede admitir múltiples dispositivos master. Cada dispositivo envía/recibe datos usando solo un cable (SDA), mientras que SCL mantiene la sincronización entre los dispositivos a través del reloj común que proporciona el master activo. Existen además algunas otras diferencias, como por ejemplo que SPI permite velocidades de acceso bastante superiores a I2C, pero utiliza mas pines de conexión.

Para utilizar este modelo de display es recomendable instalar la librería `ssd1306`, la cual incorpora comandos que facilitan la comunicación por I2C. A continuación, un ejemplo simple de uso donde se configura el protocolo I2C a través de los pines 20 y 21 de la Raspberry (SDA y SCL respectivamente) y luego se escribe el texto "TESIS" en las coordenadas (x,y)=(20,30) del display de 128x64 píxeles (ver mayor detalle en los comentarios del propio código):

```
from machine import Pin, I2C            # clase Pin e I2C de libreria machine
from ssd1306 import SSD1306_I2C         # clase SSD1306_I2C de libreria ssd1306
I2C_SDA_PIN = machine.Pin(20)           # configuracion de Pin 20 como SDA
I2C_SCL_PIN = machine.Pin(21)           # configuracion de Pin 21 como SCL
i2c = machine.I2C(0, sda=I2C_SDA_PIN, scl=I2C_SCL_PIN, freq=400000)     # I2C
oled = SSD1306_I2C(128, 64, i2c)        # display de 128x64 píxeles, I2C
oled.fill(0)                            # borra el display
oled.show()                             # envia comando anterior al display
oled.text('TESIS', 20, 30)              # imprime texto en x=20, y=30
oled.show()                             # envia comando anterior al display
```

En el proyecto en particular, se busca conocer la temperatura de operación del Skipper durante todo el tiempo, por lo que se incorporó esta información al display, al igual que la potencia que se le entrega al heater (sensado con el módulo de telemetría ya descripto). Luego, si se selecciona la opción de "control LED", se muestra la intensidad (de 0 a 100 %) en pantalla. En la Fig. 2.18 se ejemplifican algunas de esas pantallas en el display.

Figura 2.18: Ejemplo de visualización de diferentes pantallas del sistema implementado, en display OLED 1306 de 128x64 píxeles.

Capítulo 3

Fuente de alimentación y telemetría

En este capítulo se describe el desarrollo de la fuente conmutada con regulación digital, y se discuten otras alternativas exploradas hasta llegar a la versión definitiva utilizada en el dispositivo final. Se incluyen también las mediciones realizadas para caracterizar las fuentes que se exploraron.

En la Sec.3.5 se discute la implementación de un módulo de telemetría, junto con el procedimiento de calibración para las mediciones de corriente y tensión.

3.1. Control de fuente ON/OFF

Uno de los componentes fundamentales del controlador de temperatura, es la fuente de alimentación de potencia. A partir de ella se controla la tensión que llega al *heater*, y por lo tanto la temperatura del sistema. En ese sentido, originalmente se evaluó la posibilidad de utilizar una fuente de notebook (V_{out} = 19,5 V, I_{out} = 3,42 A) e implementar un circuito de encendido y apagado electrónico controlado con la Raspberry, y de esa forma controlar la temperatura a través del tiempo en un estado o en el otro. Tal circuito de control de la fuente (tipo ON-OFF) se aprecia en la Fig. 3.1.

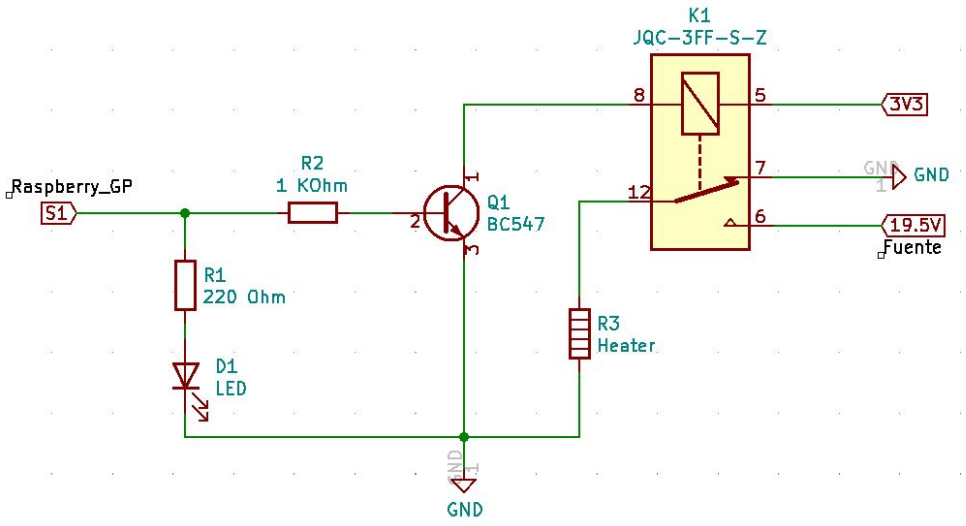

Figura 3.1: Circuito de control de heater, tipo ON-OFF, con relé.

En tal circuito, la salida S_1 es controlada por un GP de la Raspberry (el cual se puede configurar en 3,3 V - un UNO lógico a la salida - ó 0 V - un CERO lógico a la salida), la rama de R_1 y el LED D_1 sirven como testigo (al tener un UNO en S_1 se prende el LED), R_2 limita la corriente en la base del transistor Q_1, el cual funciona como interruptor: con un UNO en S_1, el transistor esta en estado de *saturación*, por lo tanto conduce corriente entre colector y emisor (llave cerrada), se acciona el relé y por lo tanto se cierra el circuito que prende el heater. Al configurar un CERO en S_1, el transistor pasa al estado de *corte*, no conduce corriente (llave abierta) y por lo tanto no se acciona el relé y en consecuencia no se prende el heater.

El motivo por el que se usa un relé como llave y no directamente el transistor Q_1, es que con el relé se pueden controlar tensiones y corrientes mayores. Por ejemplo, alimentando un relé con 5 V, uno podría controlar una lámpara conectada a la red eléctrica (220 V).

Cabe aclarar que el relé que se pensó usar es de tipo SSR (*Solid State Relay*: relé de estado sólido), ya que a diferencia de los de tipo mecánicos, éste no tiene bobinados internos y por lo tanto se evita cualquier problema que pueda traer su campo magnético y los picos del corriente al conmutar [33].

Por otro lado, este tipo de solución permite únicamente prender o apagar el heater y no da lugar a regular su potencia. Incluso consideramos que el hecho de tener tantas conmutaciones puede incorporar ruido electrónico en el sistema y afectar a las mediciones que se quieren realizar con el Skipper-CCD. Es por eso que se pensó en una solución alternativa, que permita fundamentalmente regular la potencia que se le entrega al heater, evitando esas conmutaciones. De allí nació la idea de diseñar una fuente switching (conmutada).

Vale destacar que el diseño del circuito de control de tipo ON-OFF que se describió previamente, luego fué utilizado por uno de los grupos de laboratoristas (ver capítulo 2) y también se usó para incorporar funcionalidades adicionales a la fuente, como es la de prenderla/apagarla completamente.

3.2. Fuente de alimentación conmutada

El tipo de fuentes de alimentación conmutada (*SMPS: Switching Mode Power Supply*) que se describirá en esta sección, surge de la intención de reducir el costo su fabricación. En ese sentido, siendo que el componente mas caro de una fuente convencional es el transformador, lo que se buscó fue eliminarlo. Como eso no es posible, lo que se pensó entonces fue reducir su tamaño: para lograr eso sin impactar en la potencia que entrega la fuente, se necesita elevar la frecuencia de trabajo. Es decir, si en lugar de trabajar con los 50 Hz de frecuencia de línea, se logra operar a por ejemplo 100 KHz, luego el tamaño del transformador se reduce notablemente, y consecuentemente el costo de la fuente.

En resumen, como ventajas, este tipo de fuentes son mas chicas, mas económicas y mas eficientes, pero tienen el inconveniente de ser mas complejas y de tener mayores picos de tensión al conmutar, lo que se traduce en ruido eléctrico indeseado (por esa razón es importante que cuenten con buenos filtros pasa bajos).

Para comprender el modo de funcionamiento de las fuentes conmutadas, se puede observar el esquema modular de la Fig. 3.2.

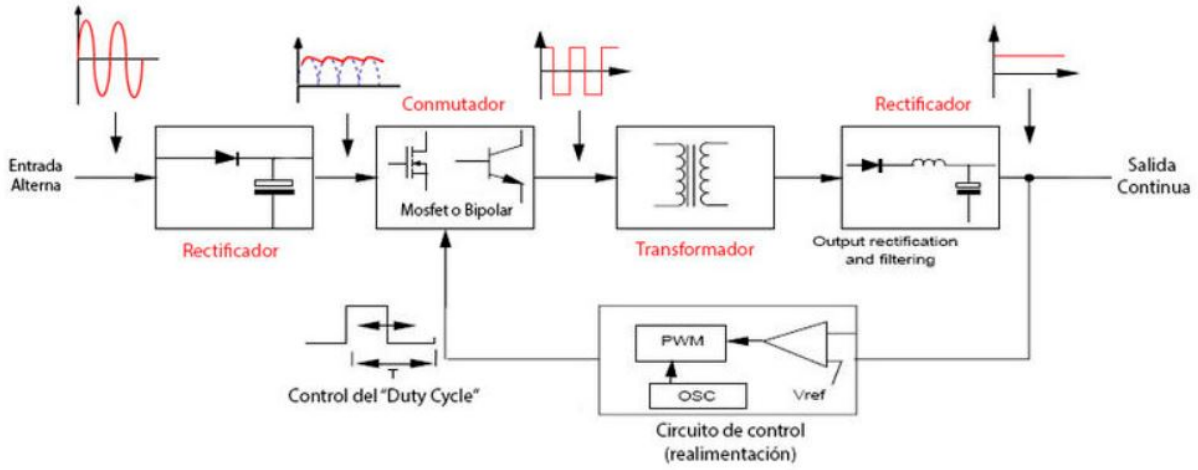

Figura 3.2: Diagrama en bloques de una fuente conmutada genérica. Se esquematizan los bloques principales: rectificador, conmutador, transformador, rectificador y el lazo de realimentación.

En la **entrada alterna** se tiene la tensión de red ($V_{RMS} = 220$ V, $V_p \simeq 310$ V, $f_L = 50$ Hz), luego el **rectificador** cumple la función convertir la entrada en tensión contínua de aprox. 310 V (generalmente se usa un puente de diodos y capacitores), a continuación el **conmutador** actúa de llave electrónica, habilitando o no el paso de corriente, por lo tanto a su salida nuevamente hay una señal alterna pero en este caso de mayor frecuencia (por ejemplo 100 KHz). La función de conmutador generalmente la cumple un transistor (típicamente un MOSFET ya que suelen ser mas rápidos y eficientes que los bipolares), donde el tiempo de encendido/apagado esta comandado por el circuito de control.

Seguido al conmutador se encuentra el **transformador**, que esencialmente reduce la tensión de la señal pulsante de 310 V a un valor mucho mas chico (por ejemplo 20 V), con la ventaja de que al tener una señal de entrada de frecuencia alta, posibilita la reducción de tamaño del transformador, como se mencionó anteriormente. Luego se encuentra otro **rectificador** y **filtro**, generalmente compuesto por un diodo, un inductor y un capacitor, generando una salida de tensión contínua.

Por otro lado, tenemos el lazo de realimentación (**circuito de control**), que tiene como componente principal un PWM (*Pulse Wave Modulator* o Modulador por Ancho de Pulso). Este componente tiene la capacidad de detectar si la carga tiene una resistencia alta o baja, y regula el ancho de un pulso consecuentemente, de manera de "prender"la llave conmutadora por mas o menos tiempo. Es decir, si por ejemplo se tiene una resistencia de carga R_L baja, se demandará mayor corriente a la fuente (mayor trabajo/potencia). Luego, para mantener la tensión de salida constante, el PWM genera un pulso de alto ciclo de trabajo y se accionará la llave del conmutador durante un tiempo mayor (ver Fig. 3.3), entregando mayor potencia a la carga. En la mencionada figura se puede observar una corriente de salida I_{OUT} mas alta, para el caso de una R_L baja, donde para ambos escenarios la tensión es la misma. Nótese también que la frecuencia para ambos tipos de carga es la misma, solo cambia el ciclo de trabajo, por eso recibe el nombre de PWM.

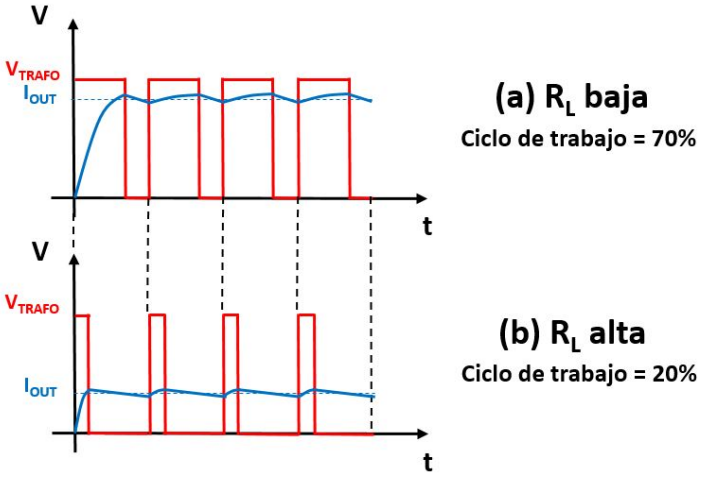

Figura 3.3: Ciclo de trabajo del PWM para una resistencia de carga R_L baja (a) y una alta (b). En azul se observa la corriente en un caso y en el otro (mayor corriente I_{OUT} a menor resistencia de carga R_L).

3.3. Diferentes opciones de fuentes conmutadas

En la sección anterior se describió el funcionamiento de las fuentes conmutadas, en forma de diagrama en bloques. Aquí se detallarán los componentes específicos que se encuentran en tal tipo de fuentes y en particular la que se utilizó para el desarrollo de la presente Tesis. También vale destacar que sólo me referiré a fuentes de tipo reductoras (*buck*). Las de tipo elevadoras (*boost*) son levemente diferentes, en especial la parte del rectificador y filtro.

En la Fig. 3.4 se esquematiza una fuente conmutada con sus componentes básicos y elementales. En ese circuito, la tensión de entrada ya fue rectificada, por lo que no se incluyó el puente de diodos y los capacitores que generan el voltage contínuo de entrada V_{cc}. Entonces, en la entrada se encuentra un transistor MOSFET (Q_1) que auspicia de llave, comandada por el pulso que recibe en el *gate* (Pin 2). Luego, de acuerdo a ese pulso, se permitirá o no el paso de corriente entre el *drain* (Pin 3) y *source* (Pin 1), lo cual se corresponde con el bloque **conmutador** que se encuentra en la Fig. 3.2 y se explicó en la previamente. A continuación se observa el diodo D_1, el inductor L_1 y capacitor C_1, que cumplen la función de rectificar y filtrar la salida de alta frecuencia del transistor, obteniendo un voltage contínuo V_{out} a la salida de la fuente.

Luego, regulando el ancho del pulso mencionado (ciclo de trabajo del transistor), se aumenta o disminuye la tensión de salida. En este ejemplo concreto, si tenemos una $V_{cc} = 30$ V, $L_1 = 10$ μH, $C_1 = 10$ μF y $R_L = 10$ KΩ, con un ciclo de trabajo del 50 % se obtiene una $V_{out} = 7{,}8$ V, y con un ciclo de trabajo del 75 %, la tensión de salida aumenta a $V_{out} = 9$ V.

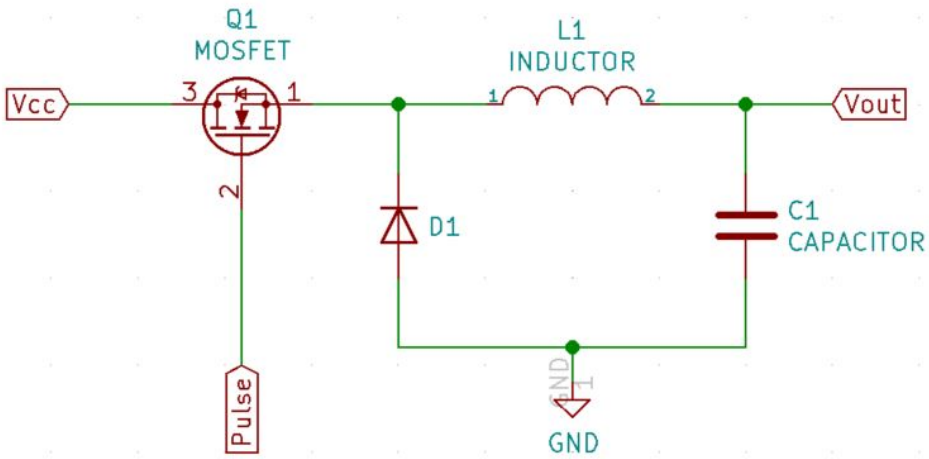

Figura 3.4: Circuito de fuente conmutada con componentes básicos.

En cuanto a la parte de filtrado en particular, lo mas simple y común que se suele encontrar, es un filtro pasabajos compuesto por un inductor y un capacitor (filtro pasivo de segundo orden). Este tipo de filtro, al no contener resistencias, tiene la ventaja de ser mas eficientes ya que casi no tienen pérdidas de energía por efecto Joule. En ese sentido, en la Fig. 3.5 se observa un ejemplo concreto de filtro mencionado, donde la frecuencia de corte se calcula como $f_o = \frac{1}{2\pi\sqrt{LC}} = 5{,}9$ KHz.

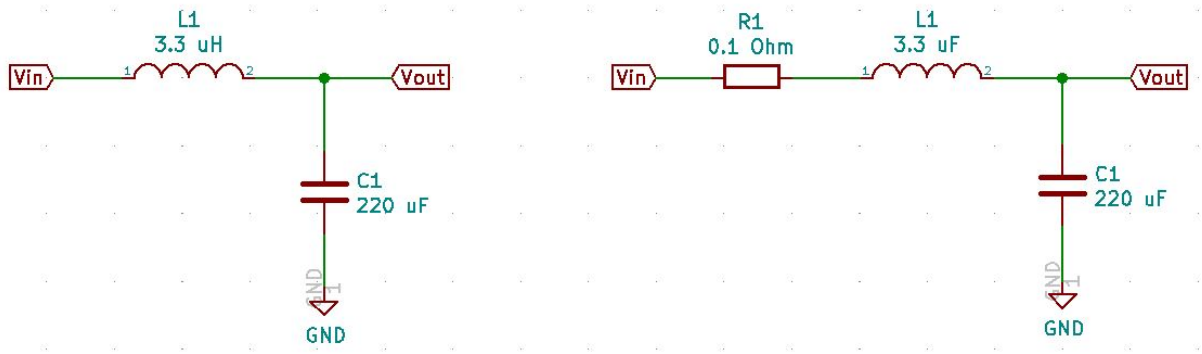

Figura 3.5: Ejemplos de filtros pasabajos de salida de fuentes conmutadas (LC y RLC) con frecuencia de corte $f_o = 5{,}9$ KHz.

Para el proyecto en si, se evaluó una primera opción de fuente conmutada reductora, basada en el LM2596 [34], cuyo circuito esquemático se encuentra en la Fig. 3.6 y se expone a modo referencial. En dicha figura se pueden ver algunos de los componentes principales ya descriptos en las fuentes conmutadas mas básicas, como son el inductor, el capacitor, el diodo rectificador y el componente de control.

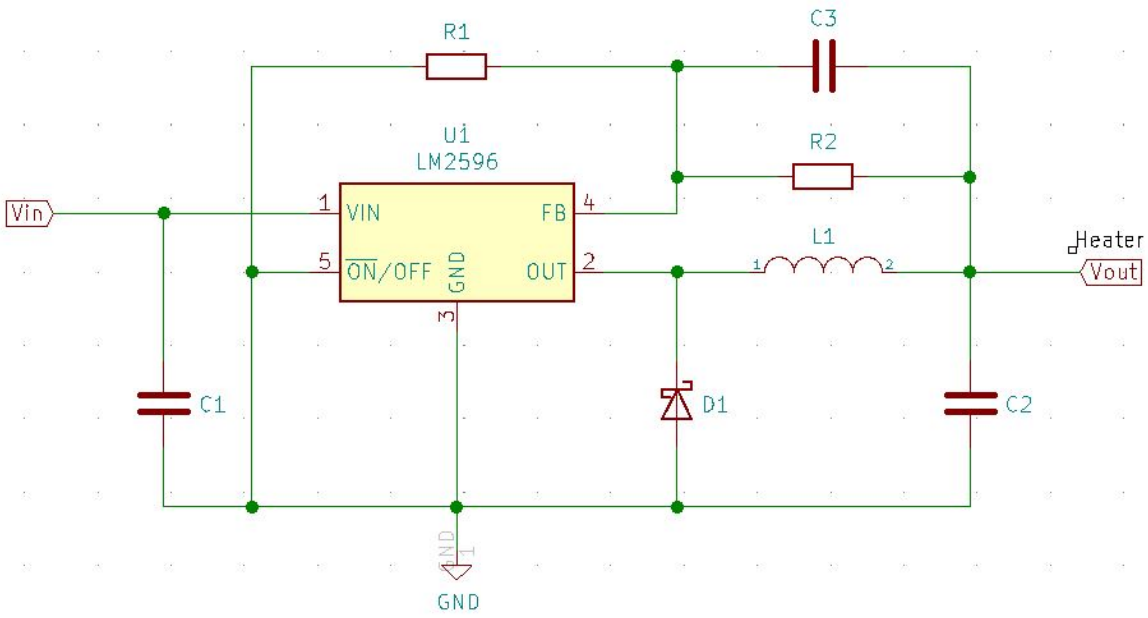

Figura 3.6: Circuito de fuente conmutada reductora basada en LM2596.

Luego, siendo que las fuentes comerciales que se consiguen de manera mas fácil en el mercado, son de V_{out} = 1,25 a 35 V, I_{out} = 3 A y f = 150 KHz, se consideró como mejor alternativa la utilización de una fuente basada en el XL4016 [29]. Dicha fuente, soporta corrientes de salida mas altas, y opera a una frecuencia superior (V_{out} = 1,25 a 35 V, I_{out} = 8 A, f = 180 KHz). Esa es la fuente que se usó para el desarrollo del proyecto, cuyo esquemático se observa en la Fig. 3.7, la cual es alimentada por un fuente comercial de notebook (V = 19,5 V, I_{max} = 3,42 A).

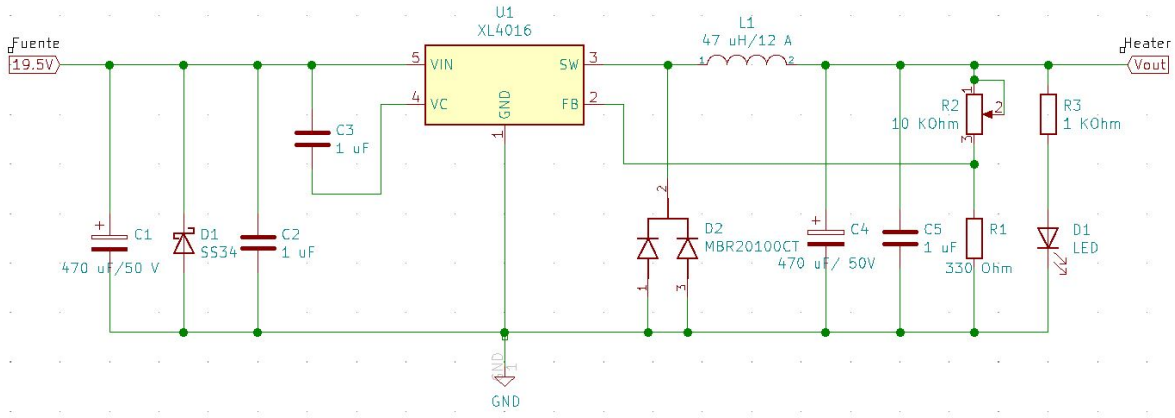

Figura 3.7: Circuito de fuente conmutada reductora basada en XL4016, utilizada como base en el diseño de la fuente del proyecto.

3.4. Control digital de la fuente conmutada

La fuente utilizada (Fig. 3.7) permite regular su tensión de salida V_{out} en forma manual, a través de un potenciómetro $R_2 = 10$ KΩ, pero lo que se buscaba es regular la potencia del sistema en forma automática. Es decir, la idea es que al medir la temperatura (con la Max31865 descripta en el capítulo 2), se pueda regular la potencia que entrega la fuente para llegar a la temperatura deseada. Por lo tanto fue necesario implementar un mecanismo de control digital, configurable por software.

A tales efectos, inicialmente se estudiaron alternativas de distintos potenciómetros digitales (MCP4131-103, X9C104, AD5252) y se eligió el MCP4131-103 [35] (7 bit, 10 KΩ), el cual permite obtener un resistencia de salida entre $\simeq 0$ y 10000 Ω, discretizada en $2^7 = 128$ pasos, es decir, $\simeq 78,125$ Ω. En rigor de verdad, la escala no comienza exactamente en 0 Ω, ni tampoco los pasos son de exactamente ese valor de resistencia. Esto se debe a que se trata de un potenciómetro digital y sus características constructivas son diferentes a las de una resistencia o potenciómetro convencional, y por lo tanto no se comporta exactamente igual en todo el rango de su escala. De todas maneras, en este caso eso no es un problema, porque lo que se busca es controlar la salida de la fuente por código y no mecánicamente, independientemente del valor que tome la resistencia que lo permita.

En primera instancia se hicieron pruebas únicamente con el MCP (sin incorporarlo al circuito de la fuente), conectándolo a la Raspberry Pi Pico como se detalla en la Tabla 3.1, y utilizando el protocolo SPI se desarrolló una rutina que permite configurar una variable `resistencia` entre 0-10 KΩ, medible con un multímentro entre los pines 6 y 7 del potenciómetro digital.

Raspberry		**MCP4131-103**	
Pin #7	SPIO CSn	*Pin #1*	$\overline{\text{CS}}$
Pin #4	SPIO SCK	*Pin #2*	SCK
Pin #6	SPIO Rx	*Pin #3*	SDI/SDO
Pin #3	GND	*Pin #4*	Vss
		Pin #5	P0A
		Pin #6	P0W
		Pin #7	P0B
Pin #36	3V3	*Pin #8*	Vdd

Tabla 3.1: Detalle de conexión de pines entre Raspberry Pi Pico y el potenciómetro digital MCP4131-103 de 10 KΩ.

A continuación se muestra parte de ese código de control, cuyo funcionamiento necesita de la instalación previa de la librería `gpio`. En dicha rutina primero se definen los pines de SPIO de la Raspberry, luego se expone una función `set_value(value)` que recibe un valor entre 0 y 127, correspondiente al nivel de resistencia entre 0 y 10K Ω que se quiere configurar en el potenciómetro digital. Para mayor detalle, ver los comentarios en el propio código.

```
0 import time                          # libreria time
1 import gpio as GPIO                  # libreria
2
3 # Configuracion de los pines de control de la Raspberry Pi Pico
```

```
SPI_CS_PIN      = 5                     # SPI0 Cs (GPIO 5, Pin 7)
SPI_CLK_PIN     = 2                     # SPI0 CLK (GPIO 2, Pin 4)
SPI_SDISDO_PIN  = 4                     # mosi = SPI0 Rx (GPIO 4, Pin6)

# Configuracion de los puertos del Raspberry usando la libreria GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setup(SPI_CS_PIN, GPIO.OUT)
GPIO.setup(SPI_CLK_PIN, GPIO.OUT)
GPIO.setup(SPI_SDISDO_PIN, GPIO.OUT)

# Funcion que configura el valor de resistencia deseado
# En "value" se pasa el valor de R que se necesita: 0 = 0 ohm, 127 = 10K Ohm
def set_value(value):
    GPIO.output(SPI_CS_PIN, True)
    GPIO.output(SPI_CLK_PIN, False)
    GPIO.output(SPI_CS_PIN, False)
    b = '{0:016b}'.format(value)
    for x in range(0, 16):
        GPIO.output(SPI_SDISDO_PIN, int(b[x]))
        GPIO.output(SPI_CLK_PIN, True)
        GPIO.output(SPI_CLK_PIN, False)
    GPIO.output(SPI_CS_PIN, True)

resistencia = 2332                      # valor de R deseada, en Ohms
level = int(resistencia*128/10000)      # convierte R a bits internos

while True:                             # loop infinito
    set_value(level)                    # llama a la funcion que configura R
    time.sleep(0.1)
```

Una vez que se comprobó su correcto funcionamiento, el siguiente paso fue integrarlo a la fuente. Entonces, se reemplazó el potenciómetro mecánico R_2 por el digital (pines 6 y 7 del MCP4131), y cambiando la variable `resistencia` del código anterior, se buscó regular la tensión de salida la fuente. Eso no funcionó debido a una limitación de la arquitectura interna del potenciómetro digital. Al investigar el problema se descubrió que lo que ocurre es que los pines FB y SW del XL4016 de la fuente (ver Fig. 3.7 y datasheet [29]), generan tensiones mayores a las que soporta el MCP para funcionar correctamente. En concreto, los pines P0A, P0W y P0B deben operar en el rango de -0,3 V a V_{DD} +0,3 V según la hoja de datos, pero nuestra $V_{DD} = 3,3$ V.

Para intentar sortear este obstáculo se invirtió de lugar R_1 y R_2 en el circuito. En esta configuración el potenciómetro digital ya no tiene voltages no soportados en sus pines. Entonces, para el circuito original, la tensión de salida se puede estimar con la Ec. 3.1, de acuerdo a la hoja de datos del XL4016.

$$V_{out} = 1{,}25 \left(1 + \frac{R_2}{R_1}\right) \tag{3.1}$$

Ahora bien, al invertir las resistencias de lugar, la V_{out} es la de la Ec. 3.2, siendo R_1 una resistencia fija y R_2 la del potenciómetro variable.

$$V_{out} = 1{,}25 \left(1 + \frac{R_1}{R_2}\right) \tag{3.2}$$

Para evaluar el comportamiento de la tensión de salida, se hicieron pruebas para distintas R_1 (0,99, 1,47, 2,02, 2,54 y 3,01 KΩ), haciendo un barrido en toda la escala de R_2 y se graficó V_{out} vs. R_2, como se observa en la Fig. 3.8. Se puede observar que el comportamiento es hiperbólico, como predice la Ec. 3.2.

Los valores de R_1 usados fueron o bien resistencias comerciales o resistencias equivalentes construidas con las comerciales y luego medidas con un multímetro. Se buscó cubrir el rango de 1 a 3 KΩ, de a 500 Ω (1, 1,5, 2, 2,5 y 3 KΩ).

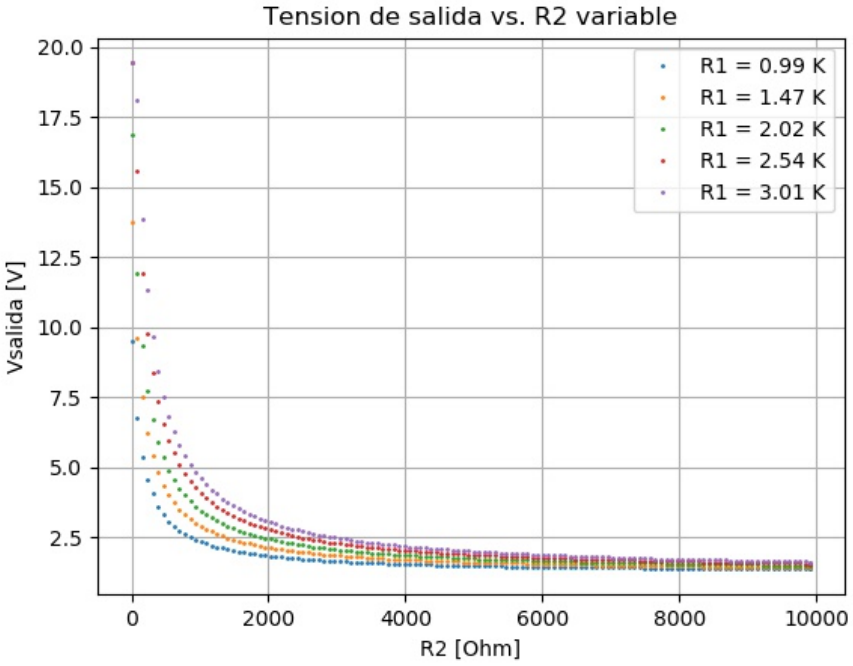

Figura 3.8: V_{out} en función del potenciómetro digital R_2 (MCP4131-103), para distintos valores de R_1 fija.

La respuesta hiperbólica de la tensión de salida tiene el inconveniente de que en la escala mas baja de R_2 los saltos de tensión son mas grandes, mientras que en el otro extremo de la escala (por ejemplo, para $R_2 > 5$ KΩ), cada paso del potenciómetro resulta en una variación muy pequeña de V_{out}. Esa situación se observa en detalle en la Fig. 3.9, donde se recortó la escala de R_2 en 1 KΩ, y se representaron los valores discretos que adopta la tensión de salida, para cada uno las mismas cinco R_1 fijas evaluadas previamente.

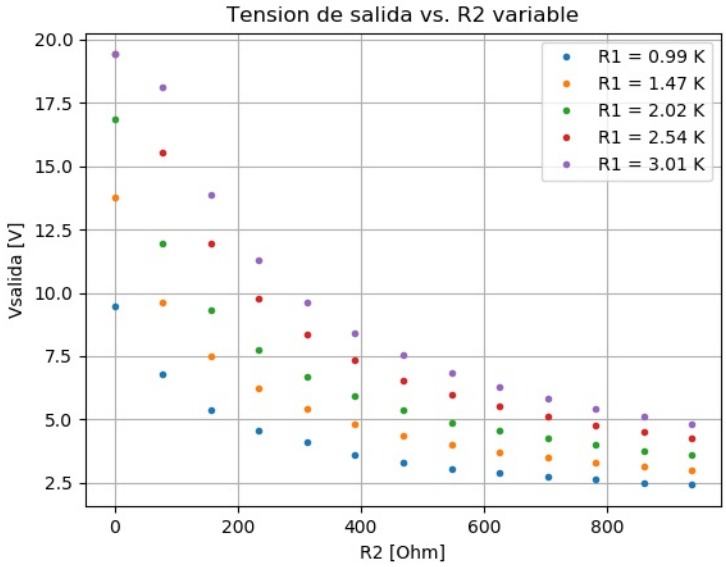

Figura 3.9: V_{out} en función del potenciómetro digital R_2 (hasta 1 KΩ), con pasos discretos, para distintos valores de R_1 fija.

Una vez caracterizada la fuente, se determinó el impacto de agregar una carga en la V_{out} para simular un heater. Con este fin se usaron dos resistencias de potencia $R_L = 8\ \Omega$ y 13 Ω, y se comparó contra la situación de no tener carga (circuito abierto), usando $R_1 = 3{,}01\ \Omega$ y barriendo R_2 hasta 400 Ω, como se aprecia en la Fig. 3.10. No se observaron diferencias significativas en la tensión de salida para los 3 casos evaluados. Como en esta medición no se registró la corriente de salida, es posible que ésta si dependiera de la carga utilizada. Las mediciones posteriores se llevaron a cabo contemplando esta situación, es decir, se midió tensión y corriente (calculando la potencia), para las distintas cargas.

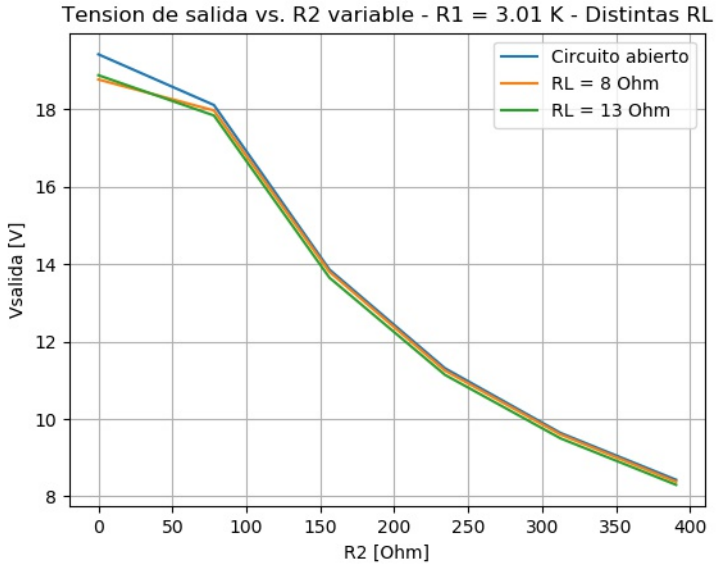

Figura 3.10: V_{out} en función del potenciómetro digital R_2 (hasta 400 Ω), para $R_1 = 3{,}01$ KΩ y distintas cargas: $R_L = 8\ \Omega$, 13 Ω y circuito abierto.

En la curvas referenciadas hasta ahora se ve que los pasos de las tensiones de salida están limitados por la discretación de R_2 (de $\simeq 78,125\ \Omega$, como se mencionó). Por ese motivo se buscó reducirlos a la mitad, colocando un segundo potenciómetro digital MCP4131 de 10 KΩ en paralelo; es decir, la resistencia máxima pasa a ser 5 KΩ y los pasos 39,062 Ω. Esta prueba apuntó a determinar si tiene sentido implementar la regulación con un potenciómetro de 5 KΩ (en lugar de 10 KΩ) y con el doble de resolución. Incluso se podría utilizar uno de 8 bits en vez de 7, lo que consecuentemente cuadruplica la resolución; es decir, entre 0 y 5 KΩ existirían 256 pasos en lugar de 128 entre 0 y 10 KΩ.

En la Fig. 3.11 se observa la comparación entre un caso (en (a) $R_2 = 10$ KΩ) y el otro (en (b) $R_2 = 5$ KΩ), para una $R_1 = 3{,}01$ KΩ y ambos casos de cargas $R_L = 8$ y 13 Ω. Se puede ver que la potencia de salida, en ambos casos es mas alta cuando la resistencia de carga R_L es menor.

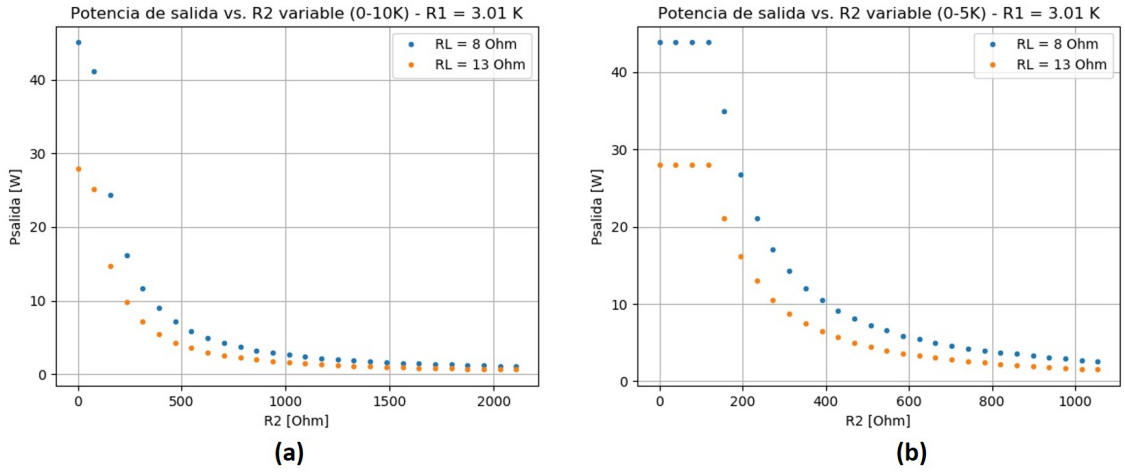

Figura 3.11: P_{out} en función del potenciómetro digital R_2, para $R_1 = 3{,}01$ KΩ y distintas cargas: $R_L = 8$ Ω y 13 Ω. (a) $R_2 = 10$ KΩ y (b) $R_2 = 5$ KΩ.

Nótese que se logra el efecto esperado, en el sentido de tener mejor resolución de potencia de salida, para el rango mas bajo de la escala de R_2. Por ese motivo se repitieron las mediciones de potencia para distintas R_1 fijas (0,99, 2,02, 3,01, 4,02 y 5,01 KΩ), como se muestra en la Fig. 3.12, en vistas de poder elegir la R_1 mas adecuada. Se observa que para las resistencias evaluadas, las curvas se van "desplazando" hacia la derecha conforme aumenta la resistencia. De todas formas, lo importante no es ese desplazamiento, sinó el hecho de que las curvas son menos pronunciadas y por lo tanto mejoran la resolución en la escala baja de R_2.

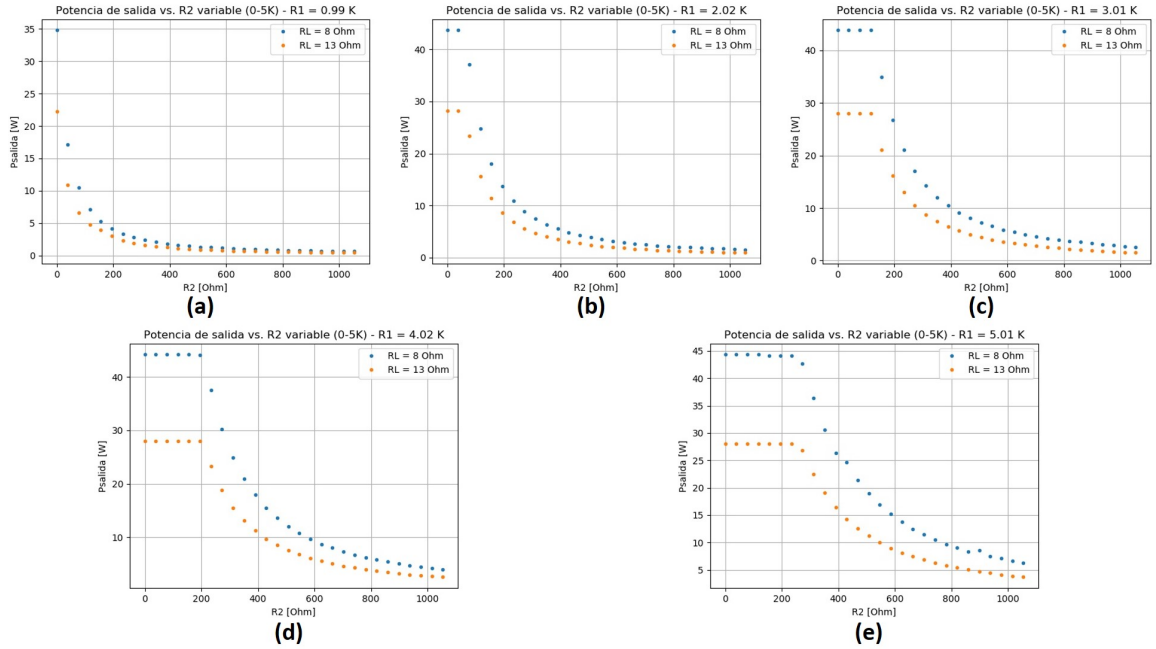

Figura 3.12: P_{out} en función del potenciómetro digital R_2 variable de 0 a 5 KΩ, para distintas R_1 fijas (0,99, 2,02, 3,01, 4,02 y 5,01 KΩ respectivamente) y cargas: $R_L = 8\ \Omega$ y 13 Ω.

En la Fig. 3.13 se busca mostrar la información anterior de manera de poder comparar los distintos escenarios y elegir las resistencias mas convenientes. Es decir, se toma el caso de R_2 variable de 0 a 5 KΩ (2 potenciómetros MCP4131 de 10 KΩ en paralelo) y se mide la potencia de salida para distintas R_1 fijas, para el caso de una carga (a) $R_L = 8\ \Omega$ y una (b) $R_L = 13\ \Omega$.

De allí se concluye que una solución viable sería utilizar $R_1 = 5$ KΩ y un potenciómetro digital de 128 pasos, variable de 0 a 5 KΩ. Utilizando la escala a partir de los 200 Ω se obtiene una buena resolución de potencia a la salida de la fuente, para cargas $R_L = 8$ y 13 Ω. Ese circuito integrado puede ser el **MCP4131-502** (7 bit, 128 pasos, 5 KΩ) o el **MCP4151-502** (8 bit, 256 pasos, 5 KΩ), que tiene el doble de resolución.

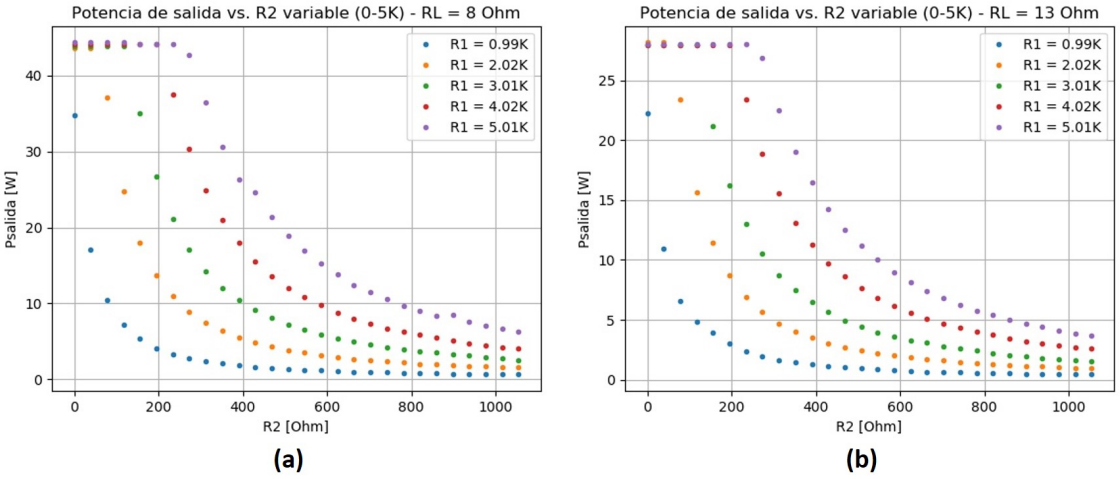

Figura 3.13: P_{out} en función del potenciómetro digital R_2 variable de 0 a 5 KΩ, para distintas R_1 fijas (0,99, 2,02, 3,01, 4,02 y 5,01 KΩ) y cargas: (a) $R_L = 8\ \Omega$ y (b) $R_L = 13\ \Omega$.

Por completitud en la Fig. 3.14 se muestra otra de las pruebas realizadas, que consistió en colocar una resistencia $R_p = 500\ \Omega$ en paralelo al potenciómetro variable $R_2 = 10\ \text{K}\Omega$, $R_1 = 3{,}01$ KΩ, con el objeto de mejorar la resolución en la escala baja. En la figura se representa el caso sin R_p ($R_L = 8\ \Omega$ en azul y $R_L = 13\ \Omega$ en naranja), y el caso con R_p ($R_L = 8\ \Omega$ en verde). No se observan mejoras significativas, por lo que no se continuó por este camino.

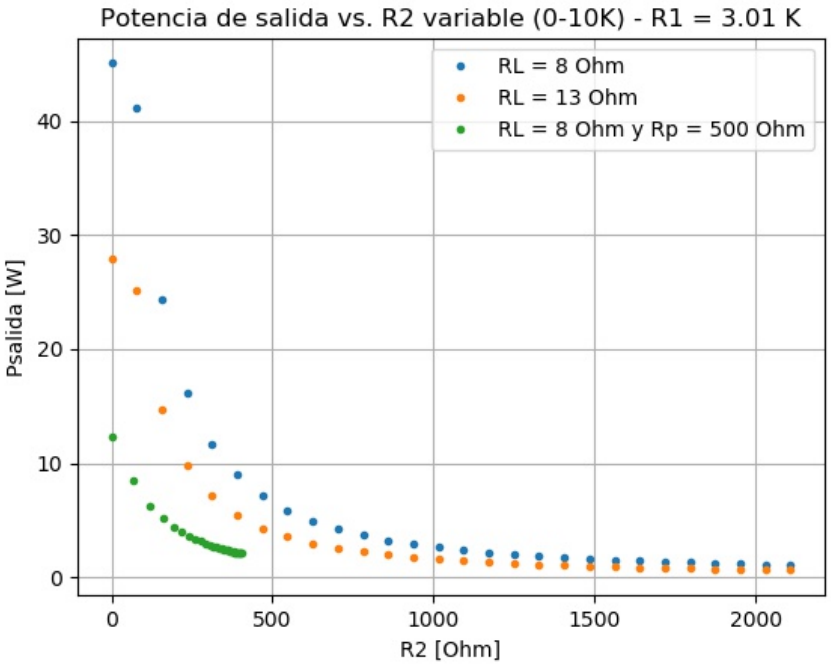

Figura 3.14: P_{out} en función del potenciómetro digital R_2 variable de 0 a 10 KΩ (MCP4131-103), con $R_p = 500\ \Omega$ en paralelo ($R_L = 8\ \Omega$) y sin R_p, para $R_1 = 3{,}01$ KΩ y cargas $R_L = 8$ y $13\ \Omega$.

Si bien podríamos haber utilizado los potenciómetros digitales anteriores, haciendo el cambio de R_1 por R_2 descripto, se decidió evaluar otro modelo de la misma familia que tiene la particularidad de ser HV (*High Voltage* - alto voltaje). Es decir que sus pines P0A, P0W y P0B ya no estan limitados a una tensión máxima relacionada con V_{DD}, sinó que ese voltage depende de otro pin, denominado V_+.

El circuito integrado en cuestión es el **MCP41HV51-502** [30] (8 bit, 5 KΩ, HV), que a diferencia de los mencionados antes, tiene funcionalidades adicionales, como ser el $\overline{\text{SHDN}}$, que permite prenderlo o apagarlo. Por ese motivo tiene 14 pines y los anteriores tienen 8, lo cual agregó una complejidad adicional que no tiene que ver exactamente con la cantidad de pines sino que estos integrados vienen en formato TSSOP-14 y los anteriores en DIP-8. Eso los hace mucho mas pequeños y mas difíciles de manipular; incluso hubo que comprar un adaptador TSSOP-DIP y soldar el integrado manualmente, lo cual no resulta fácil.

Se lo conectó a la Raspberry según se muestra en la Tabla 3.2, y el pin 14 (V_+) al positivo de la fuente de laptop (19,5 V); eso es lo que permite que el potenciómetro funcione en HV y por lo tanto se pueda usar en nuestro circuito original (Fig. 3.7), como R_2 variable. Luego, los pines 11 y 12 (P0B y P0W) son los que reemplazan dicha resistencia.

Raspberry		**MCP41HV51-502**	
Pin #36	3V3	*Pin #1*	VL
Pin #9	SPIO SCK	*Pin #2*	SCK
Pin #12	SPIO CSn	*Pin #3*	$\overline{\text{CS}}$
Pin #11	SPIO Rx	*Pin #4*	SDI
Pin #10	SPIO Tx	*Pin #5*	SDO
Pin #33	GND	*Pin #6*	$\overline{\text{WLAT}}$
Pin #1	GP0	*Pin #7*	$\overline{\text{SHDN}}$
Pin #33	GND	*Pin #8*	NC
Pin #33	GND	*Pin #9*	DGND
Pin #33	GND	*Pin #10*	V-
		Pin #11	P0B
		Pin #12	P0W
		Pin #13	P0A
		Pin #14	V+

Tabla 3.2: Detalle de conexionado entre pines de Raspberry Pi Pico y el potenciómetro digital MCP41HV51-502 de 5 KΩ.

Entonces, siendo que se vuelve al diseño original donde no se invirtieron las resistencias R_2 y R_1, luego la ecuación de evolución de la tensión de salida de la fuente es la anterior Ec. 3.1: $V_{out} = 1{,}25\left(1 + \frac{R_2}{R_1}\right)$, donde R_2 es variable (potenciómetro digital) y R_1 fija.

Con esa nueva configuración se midió la potencia de salida vs. R_2 variable, para dos cargas, R_L = 8 y 13 Ω, con R_1 = 330 Ω, como se ve en la Fig. 3.15. Allí se puede observar que el comportamiento de la potencia se corresponde con la ecuación citada, pero lo mas importante es que se logra la granularidad deseada en la P_{out} que le llega al heater, y consecuentemente un mejor control de la temperatura.

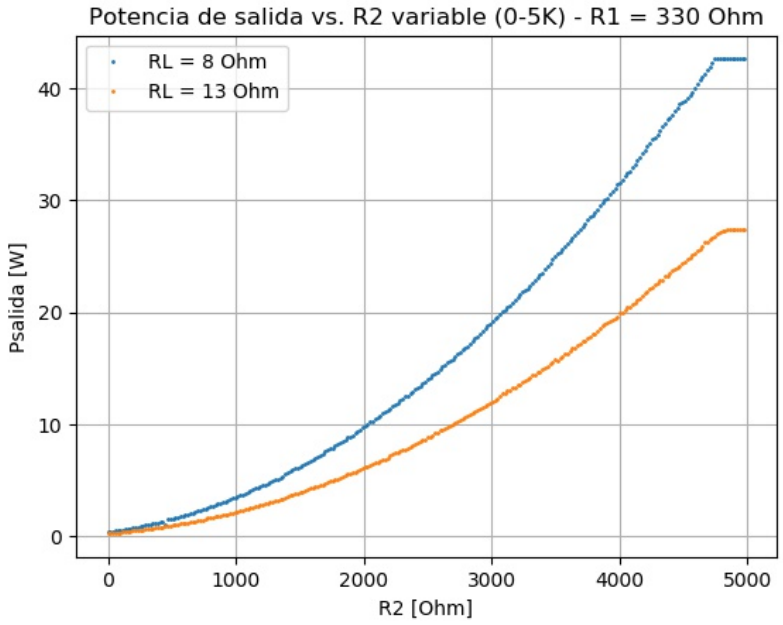

Figura 3.15: P_{out} en función del potenciómetro digital R_2 variable de 0 a 5 KΩ (MCP41HV51-502) con R_1 = 330 Ω, evaluado para dos cargas R_L = 8 y 13 Ω.

Luego se hizo otra prueba mas, para definir si se mejora la situación lograda previamente. Esa prueba consistió en utilizar una resistencia $R_1 = 165\ \Omega$ (la mitad del valor anterior, con dos R de 330 Ω en paralelo), y medir la potencia de salida al variar R_2, para una carga $R_L = 13\ \Omega$, como muestra la Fig. 3.16. No se observan mejoras sustanciales para es carga, ya que se llega a la potencia máxima antes de recorrer la escala completa de R_2 (en aprox. 2,4 KΩ), por lo que se pierde resolución de potencia.

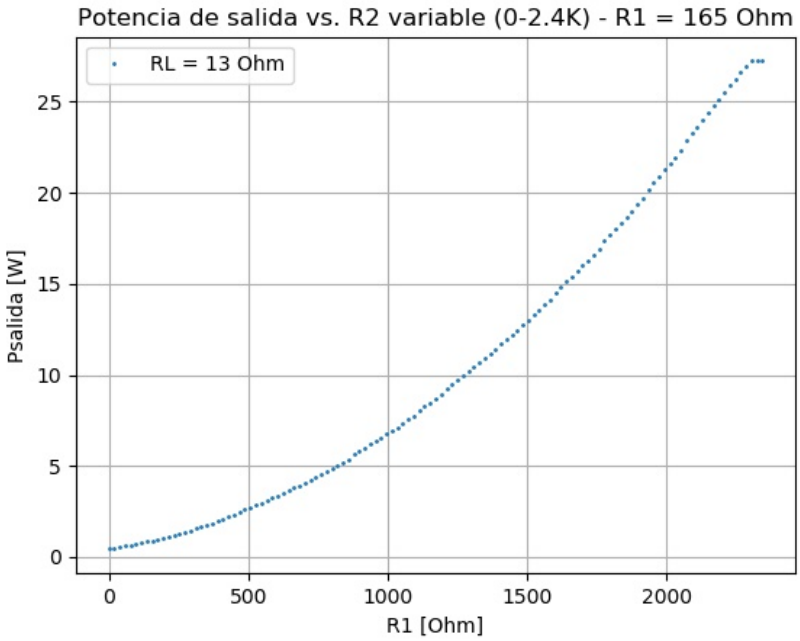

Figura 3.16: P_{out} en función del potenciómetro digital R_2 variable de 0 a 5 KΩ (MCP41HV51-502) con $R_1 = 165\ \Omega$ y carga $R_L = 13\ \Omega$.

3.5. Implementación de telemetría

La información de telemetría resulta importante en el presente sistema de control, ya que proporciona un mecanismo para conocer la potencia entregada por la fuente, en tiempo real, sin la necesidad de incorporar instrumentos externos. De esa forma se puede monitorear el funcionamiento de la fuente y obtener información de lo que esta sucediendo con el heater dentro de la cámara de vacío. En ese sentido, cabe señalar que dicha funcionalidad resultó de utilidad en gran parte de las mediciones realizadas y descriptas en este documento.

La medición de tensión y corriente para el cálculo de la potencia entregada por la fuente, se basa en la utilización de los conversores analógico-digital con los que cuenta la Raspberry Pi Pico. Estos conversores ADC tienen una resolución de 12-bit, lo que implica que podrían transformar una señal analógica en una digital en el rango de 0 a 4095 ($2^{12} = 4096$), sin embargo Microphyton lo transforma a un número de 16-bit, lo que finalmente permite ampliar el rango de 0 a 65535 ($2^{16} = 65536$).

A continuación se reproduce a modo de ejemplo, un extracto de código que permite configurar un puerto como ADC, leerlo e imprimir el valor de tensión en dicho pin:

```
from machine import Pin, ADC             # clase Pin y ADC de libreria machine
adc0 = ADC(Pin(26))                      # configuracion de Pin 26 como ADC
valor_adc0 = adc0.read_u16()             # lee ADC0
tension_adc0 = valor_adc0 * 3.3 / 65535 # convierte ADC0 a Tension (en Volts)
print (tension_adc0)                     # imprime Tension medida con ADC0
```

Por otra parte, los puertos del RP2040 soportan una tensión máxima de 3,3 V, entonces si el voltage que se quiere digitalizar esta por encima de ese valor, hará falta atenuarlo previamente de manera de no quemar el puerto. Para lograr eso se utiliza un divisor resistivo, cuya relación de conversión sea tal que la tensión analógica máxima a medir no supere los 3,3 V luego de la conversión. En concreto, para medir por ejemplo $V_{in} = 20$ V con $V_{out} < 3{,}3$ V, se necesita un factor de conversión $FC = \frac{R_2}{R_1+R_2} \simeq 0{,}165$ ($V_{out} = V_{in}FC$), que en ese sentido determina los valores de las resistencias que conforman el divisor resistivo (ver Fig. 3.17).

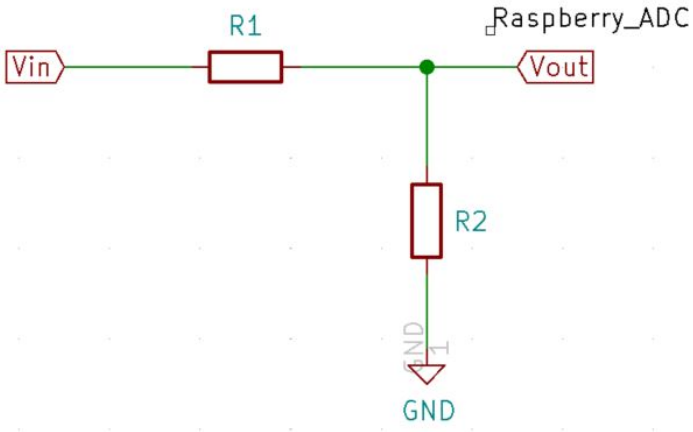

Figura 3.17: Divisor resistivo para atenuar la señal de entrada V_{in} a medir con puertos ADC de Raspberry Pi Pico. La ecuación de atenuación es: $V_{out} = V_{in}\frac{R_2}{R_1+R_2}$.

Resulta importante destacar que los conversores analógico-digital de este tipo de microcontroladores no son suficientemente precisos y confiables, al menos para esta aplicación. Específicamente lo que ocurre es que son muy sensibles a la carga que pretenden medir, por ejemplo: dependiendo de la resistencia equivalente que tenga lo que se conecte al ADC en relación a la impedancia de entrada del puerto, la digitalización funcionará mejor o peor (incluso puede ser mas precisa en parte de la escala, y menos precisa en otra parte). También ocurre que las mediciones "oscilan", y no se obtiene un valor constante convertido.

Algunos de esos problemas pueden reducirse, por ejemplo eligiendo las resistencias del divisor de tensión mas adecuadas, agregando capacitores a la entrada de los puertos ADC o utilizando como tierra del circuito a pin 33 de la Raspberry (AGND) y no cualquier GND. Sin embargo, a pesar de usar esas recomendaciones, lo ideal es generar un mecanismo de calibración que convierta los valores medidos con el ADC, con los obtenidos utilizando un multímetro. En la siguiente sección se detalla dicho mecanismo, junto con las recomendaciones implementadas para mejorar las mediciones.

3.6. Calibración

El proceso de calibración para lo que tiene que ver con la medición de tensión y corriente con Raspberry, nace del hecho de que típicamente los conversores analógico-digital (ADC) de este tipo de microcontroladores, no son demasiado precisos. En concreto en este caso, las lecturas de los ADC oscilan y en otros casos, difieren de los valores medidos y contrastados con un multímetro (por exceso en parte de la escala, y por defecto en otra).

Por el lado de las oscilaciones, hay algunas recomendaciones que mejoran la situación. Estas son: utilizar capacitores pequeños entre los pines del ADC y tierra (100 nF), utilizar ADC_VREF (pin 35) como pin de alimentación de los dispositivos de 3,3 V (en lugar de 3V3(out) - pin 36), y que la tierra del circuito sea AGND (pin 33) y no cualquier GND de la Raspberry.

Por otra parte, en cuanto a la medición de tensión, la fuente diseñada es capaz de entregar hasta un máximo aproximado de 18,5 V. En rigor de verdad, podría entregar mayor tensión y corriente, pero esa limitación esta impuesta por la fuente de notebook utilizada (19,5 V/ 3,42 A). Es decir, la fuente BUCK no puede entregar mayor potencia de la que recibe como entrada.

Ocurre que a pesar de que V_{out} máxima es 18,5 V aprox., el voltage máximo soportado por los puertos de la Raspberry es 3,3 V. Es por eso que fue necesaria la incorporación de un divisor resistivo. Dicho divisor no solo tiene que ser capaz de reducir la tensión a medir sin superar el máximo soportado (3,3 V), sinó que también debe tener una resistencia equivalente lo suficientemente alta como para no cargar al heater (cuya resistencia ronda los 10 Ω), ni alterar la resistencia de entrada del puerto ADC. De esa forma, se utilizaron resistencias de 10 KΩ y 2 KΩ, lo que da un factor de conversión $V_{out} = V_{in} \frac{2K\Omega}{2K\Omega+10K\Omega} = V_{in}$ 0,166. Luego, si V_{in} = 18,5 V $\Rightarrow V_{out}$ = 3,08 V. En la Fig. 3.18 se observa el circuito implementado para obtener la telemetría. En él se aprecia el divisor de tensión y los capacitores de 100 nF mencionados antes, junto con el sensor de corriente y la parte de calibración que se describe a continuación.

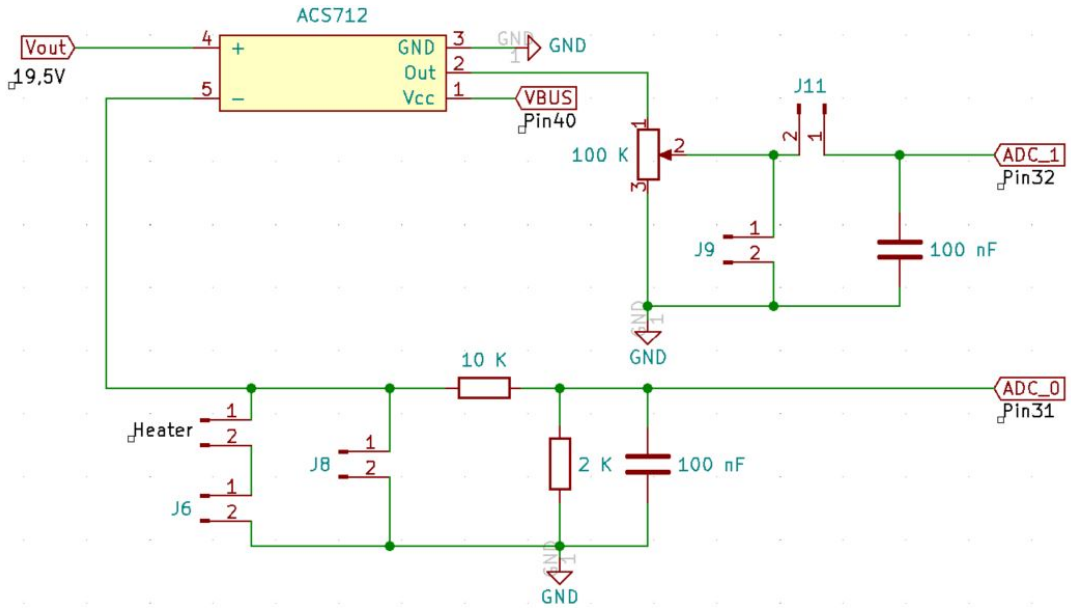

Figura 3.18: Circuito implementado para medir tensión y corriente con los ADC de la Raspberry Pi Pico.

En el caso de la medición de corriente, se resuelve con un método indirecto, siendo que los puertos ADC convierten cierta tensión analógica, a un valor digital (entre 0 y 65535). Entonces, para "traducir"tensión a corriente, se usó un sensor de por efecto Hall (ACS712 de 5 Ampere). Dicho sensor se alimenta con 5 V (no con 3,3 V) y entrega un voltage proporcional a la corriente que circula, a partir del punto medio de dicha tensión de alimentación. Es decir, como $V_{cc} = 5$ V, para corrientes positivas el dispositivo entrega 185 mV/A al estar por encima de 2,5 V ($V_{cc}/2$) y el mismo factor de conversión para corrientes que circulen en dirección contraria y por lo tanto la tensión proporcional se encuentre por debajo de 2,5 V.

De esta forma, para la medición de corriente también existe la limitación de no poder superar la tensión máxima del puerto ADC, sumado al hecho de tener que sintonizar el cero de corriente con la mitad de la escala del sensor (0 A equivale a $V_{cc}/2 = 2{,}5$ V. En nuestro 0 A corresponde a $3{,}3/2 = 1{,}65$ V, para no quemar el puerto). Por ese motivo, se resolvió a través de un potenciómetro de 100 KΩ, el cual se ajusta para tener 1,65 V (mitad de la escala), cuando no circula corriente.

Luego, por los motivos detallados previamente, se decidió elaborar un mecanismo de calibración que permita tener una correspondencia entre las mediciones de los ADC (tensión y corriente), y las mediciones mediante un multímetro (ver esquemático en la Fig. 3.18). En ese sentido se desarrolló un código al cual se le puede indicar la cantidad de puntos de la escala a evaluar, entre 0 y 255 (que son los pasos del potenciómentro digital), en potencias de 2 (para tener puntos equidistantes), y la cantidad de mediciones deseadas sobre cada punto. Luego esas mediciones se promedian y se compara contra las mediciones de tensión y corriente realizadas con un multímetro, ajustando por un polinomio del grado deseado. En la Fig. 3.19 se muestra un ejemplo cualitativo para la calibración de corriente con 8 puntos equiespaciados en toda la escala de variación del potenciómetro, ajustado por una función cuadrática (aunque finalmente se adoptó un modelo lineal).

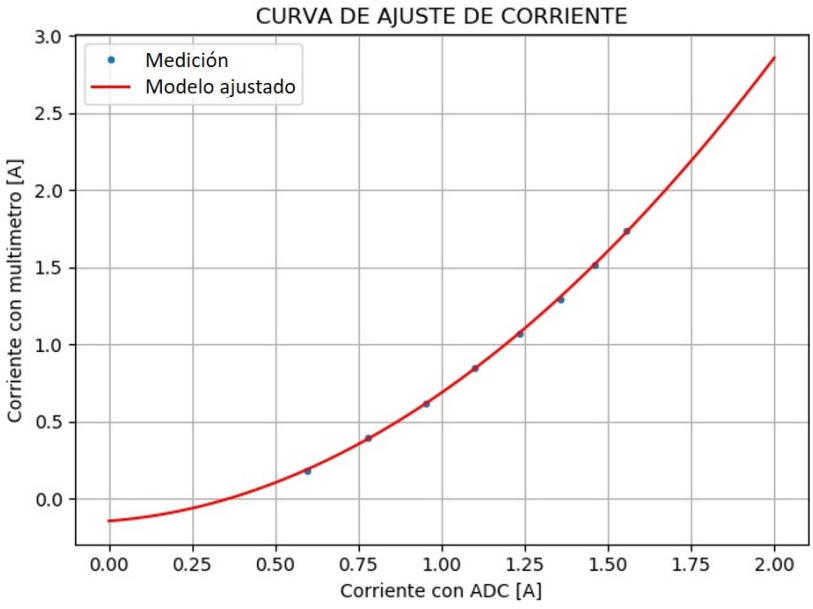

Figura 3.19: Corriente medida con multímetro vs. medida con ADC de Raspberrry, ajustado por función cuadrática.

Se observó que la función lineal es un buen ajuste (Fig. 3.20) tanto para tensión como para corriente, por lo que a futuro, sólo sería necesario sensar 2 puntos para obtener una curva de calibración. Sin embargo, como se mencionó, el código permite ajustar con polinomios del grado que se quiera, mejorando de esa forma la precisión del modelo. Finalmente, dicha rutina devuelve los parámetros de los polinomios (para tensión y corriente) que luego se configuran en el programa de funcionamiento general del proyecto. En la Fig. 3.20 se exponen las curvas de calibración te tensión (a) y corriente (b), con sus correspondientes ajustes lineales, para 8 puntos de medición y 200 muestras promediadas por punto. En el caso de la curva de tensión, los parámetros del polinómio que devuelve el código son $p_0 = 0{,}99$ y $p_1 = -0{,}16$, y para el caso de la corriente, $p_0 = 1{,}61$ y $p_1 = 0{,}19$.

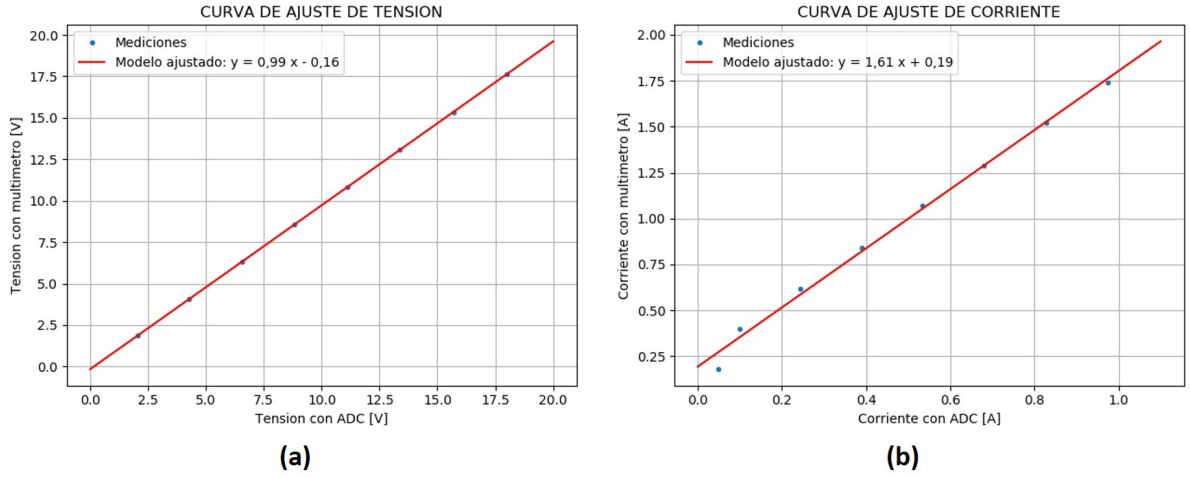

Figura 3.20: Curvas de calibración: (a) Tensión medida con multímetro vs. medida con ADC de Raspberrry. (b) Corriente medida con multímetro vs. medida con ADC de Raspberrry. Ambos casos fueron ajustados con una función lineal.

A continuación de enumeran los pasos a seguir para realizar la calibración, basado en la nomenclatura utilizada en la Fig. 3.18, con el heater enchufado:

1. Colocar jumper **J6**, retirar **J11** y medir tensión en **J9** hasta obtener 1,60 V a partir de la regulación del potenciómetro de 100 KΩ (con la precaución de medir tensión con la polaridad correcta). Volver a colocar **J11**.

2. Retirar jumper **J6** y colocar multímetro en modo corriente entre sus terminales (Precaución: validar que el multímetro pueda medir corrientes del orden de 5 A).

3. Colocar multímetro en modo tensión (max. 20 V) entre terminales **J8**.

4. Ejecutar código de calibración, habiendo previamente configurado la cantidad de muestras N y cantidad de puntos M (potencias de 2, max. 256).

5. Registrar los valores de tensión y corriente obtenidos con los multímetros para cada uno de los puntos M definidos, y los devueltos por el código a partir de las mediciones con ADCs.

6. Cargar dichos valores en el código de ajuste, el cual devuelve los parámetros p_n de los polinomios del grado definido (Ej: $y = p_0x^2 + p_1x + p_2$).

7. Ingresar esos parámetros en el programa general del proyecto.

Por último, vale destacar que el proceso de medición y calibración descripto en esta sección, se basó fuertemente en un mecanismo iterativo de prueba y error. Por ejemplo, según la documentación, los capacitores cerámicos utilizados en los puertos ADC para reducir las oscilaciones, se recomienda que sean del orden de 10 nF. Sin embargo, luego de varias pruebas, se encontraron mejores resultados con capacitores de 100 nF. De la misma forma, la utilización de ADC_VREF como fuente de 3,3 V, en este caso no mejoró las oscilaciones de los ADC, sino que por el contrario, la empeoró. Por lo tanto, se volvió a usar la salida 3V3(out) de la Raspberry. Por el lado de los divisores resistivos, también se probaron combinaciones de resistencias mas chicas y mas grandes, llegando a un mejor resultado con los valores detallados previamente. También se optimizaron las mediciones obtenidas promediando los valores de los ADC con 200 muestras; con valores mas chicos no se lograban suficientes mejoras y con valores mas altos las mejoras no resultaron significativas.

Capítulo 4

Integración de hardware y software

En este capítulo se detalla el proceso de diseño y armado del circuito explicado en el capítulo anterior. Se describen prototipos previos a la versión definitiva de la implementación final diseñada en KiCAD de la placa PCB construída, que incorpora todos los módulos descriptos anteriormente. También se explican las rutinas de control de dichos módulos.

4.1. Construcción del PCB

Los distintos módulos que componen el sistema completo se implementaron en un *protobard*, para hacer pruebas del *hardware* sin tener que soldar los componentes o los cables que los interconectan. El camino cronológico que se siguió, comenzó con la programación asociada al manejo de los distintos puertos de la Raspberry Pi Pico, que ya fueron mencionados en capítulos anteriores. Luego, se incorporaron funcionalidades adicionales mas complejas, como ser el manejo de la medición de temperatura con MAX31865 y la implementación de los circuitos de control tipo ON/OFF a través de relé que dió comienzo al diseño de la fuente switching regulada que provee la potencia del heater. Se probaron diferentes potenciómetros digitales, tomando mediciones de la tensión/corriente de salida en cada caso, hasta llegar a la mejor solución. Luego, se siguió con la implementación de la telemetría y manejo de display OLED, y la posibilidad de controlar y comandar al sistema en forma remota por puerto serial.

En la Fig. 4.1 se expone una foto del protoboard con la implementación de algunos de los módulos descriptos, en pleno proceso de desarrollo.

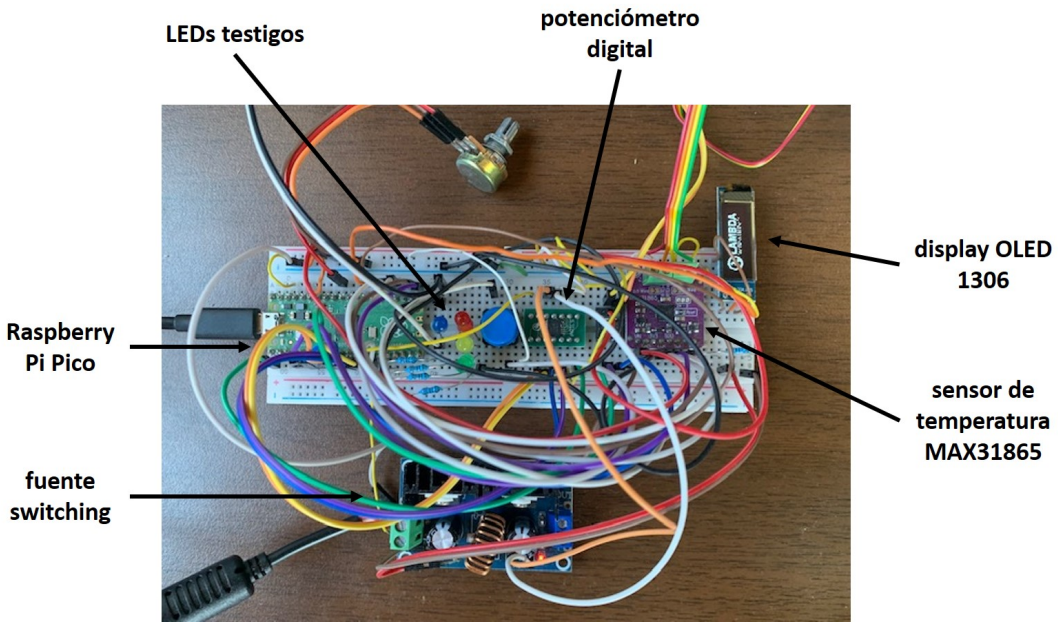

Figura 4.1: Protoboard de diseño con algunos de los circuitos implementados.

Si bien el uso de un protoboard agiliza las pruebas, tiene la desventaja de que al usar piezas móviles se corre el riesgo de tener falsos contactos o incluso se desconecte algún cable, y provocar un malfuncionamiento. Por ese motivo, una vez definido, al menos en parte, cual iba a ser el conexionado físico entre componentes y corroborado su correcto funcionamiento, se construyó el circuito en una placa de cobre perforada. Allí los componentes van soldados, al igual que los cables de interconexión. Los componentes principales (Raspberry, Max31865, display y potenciómetro digital) se colocaron en zócalos, de manera tal de poder reemplazarlos fácilmente si alguno se quemase o tuviese algún problema. La placa se muestra en la Fig. 4.2.

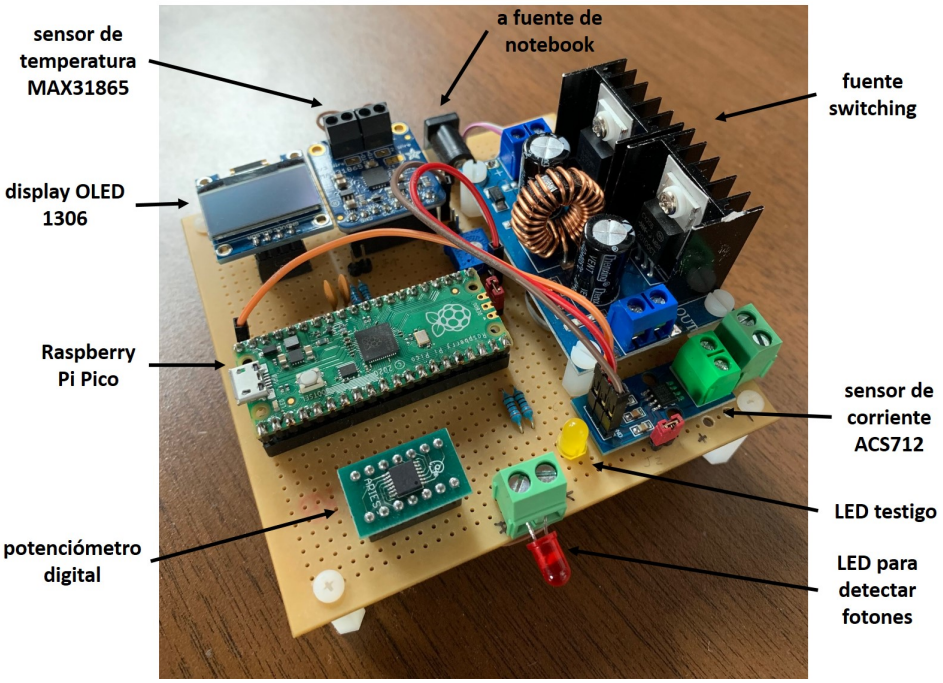

Figura 4.2: Circuito preliminar implementado en placa de cobre perforada.

Para diseñar la placa PCB utilizada en el sistema definitivo se uso el software KiCAD. El proceso inició con la implementación del circuito optimizado en los prototipos dentro del módulo de diseño de esquemáticos de KiCAD. En este proceso se utilizaron librerías de componentes ya definidos y, adicionalmente, hubo que crear los símbolos y *footprints* de los componentes no existentes (Raspberry Pi Pico, Max31865, fuente XL4016, display OLED 1306 y sensor ACS712). El circuito final se expone en la Fig. 4.3, donde se puede ver como componente central a la Raspberry **U4**, quien comanda todos los otro módulos.

Por el lado de la fuente de alimentación, se incorporaron dos mecanismos de protección: un fusible **F1** de 5 A, para evitar que se dañe el circuito si la fuente de notebook **J1** entrega mayor corriente, y un diodo Schottky **D1**, que tiene la función de proteger a la fuente frente a la posibilidad de conectar la alimentación al revés. Este tipo de diodos tienen la ventaja de tener mayor velocidad de conmutación (a diferencia de convencionales) y por otro lado tienen una caída de tensión en directa mas baja (del orden de 0,2 - 0,4 V dependiendo de la corriente, frente a $\simeq$0,7 V de un diodo común) y de esa forma "desperdician" menos potencia transferible a la carga.

Los pines 19 y 20 de la Raspberry controlan un LED testigo **D2** y el LED dentro de la cámara de vacío. El primero de ellos se agregó para indicar cuando el circuito general esta funcionando y el segundo es quien generará los fotones que se quieren detectar. Para ello se colocó una bornera de forma tal de poder conectarlo fácilmente.

Los pines 2 y 29 controlan el relé de válvula y heater respectivamente. Ambos cuentan con un LED testigo y con borneras para conectar esos dos componentes. Se observa una pequeña diferencia de diseño entre uno y otro (puntualmente el LED testigo) lo cual tiene que ver con que un circuito es normal abierto y el otro normal cerrado. Es decir, por ejemplo el heater es normal cerrado, es decir que al poner un CERO lógico en el pin 29 (GPIO_22) el relé esta cerrado y por lo tanto se prende el heater, lo cual es indicado por el LED **D4**. Lo opuesto ocurre con la válvula: se activa con un UNO en el pin 2 (GPIO_01) y se prende el LED **D3**.

Los circuitos de telemetría para la tensión y corriente están conectados a los pines 31 y 32 respectivamente (GPIO_26 ADC_0 y GPIO_27 ADC_1). La calibración de la medición de corriente se realiza ajustando la resistencia variable **RV1**. La tensión utiliza el divisor que contiene a **R5** y **R6**. Específicamente utilizan los pines y jumpers **J9** y **J11** para calibrar la corriente y **J6** y **J8** para la tensión. A su vez, se pueden ver los capacitores **C1** y **C2** encargados de reducir las oscilaciones de los ADC_0 y ADC_1 ya explicado en el capítulo 3.

El display **U6** es controlado con los pines 26 y 27 (GPIO_20 y GPIO_21). Finalmente, los pines 9, 10, 11 y 12 (GPIO_06 a GPIO_09) controlan el potenciómetro digital **U1**, que esta conectado a la fuente **U2** a través de **P1** y **P2**.

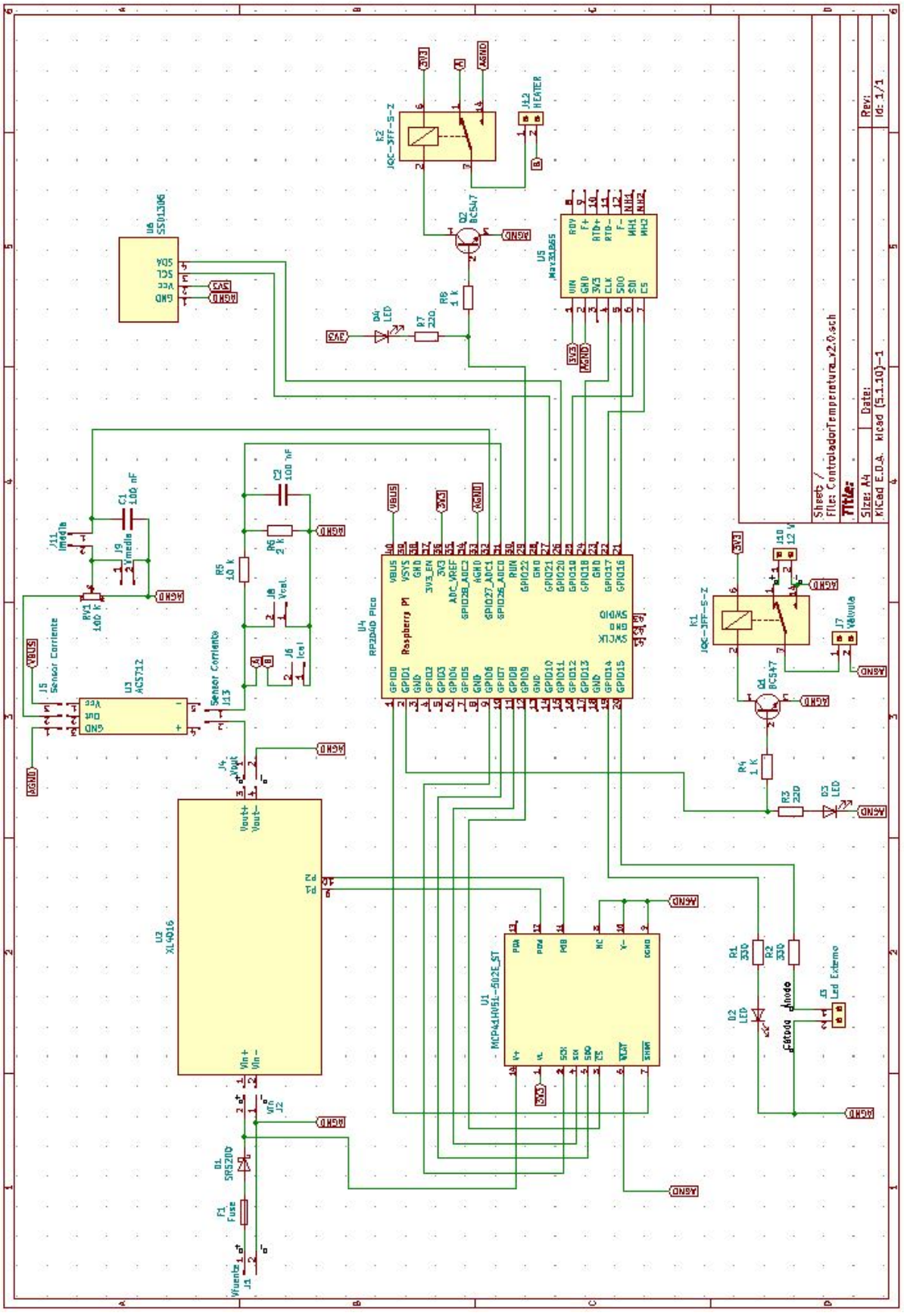

Figura 4.3: Circuito esquemático del sistema construido completo.

Luego del circuito esquemático se hizo el diseño (también en KiCAD) de la placa PCB. Para ello resultó bastante útil haber armado previamente el circuito en la placa de cobre perforada, ya que eso ayudó fundamentalmente a ubicar los componentes de manera conveniente para facilitar sus interconexiones. El PCB tiene dimensiones aproximadas de 13 x 13 cm y se realizó en dos capas, siguiendo la premisa anterior de poder conectar/desconectar los componentes principales, para reemplazarlos en caso de que alguno resulte dañado. También se colocaron las borneras de forma tal de poder acceder desde el exterior de la placa y sus LED testigos cerca de cada una de ellas. También se engrosaron aquellas pistas que manejan mayor corriente y se colocaron agujeros para poder montar la placa en un gabinete. En la Fig. 4.4 se puede ver un esquema del PCB hecho en KiCAD.

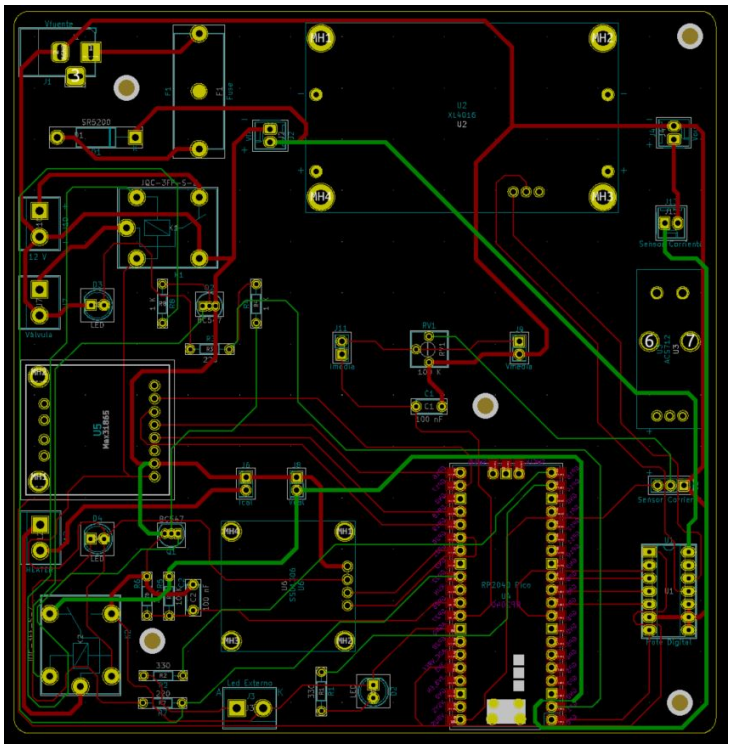

Figura 4.4: Placa PCB del sistema completo diseñada en KiCAD.

El programa utilizado para el diseño del PCB permite generar las *gerber files*, que son los archivos de salida que permiten mandar a fabricar la placa en si. Eso se realizo en una fabrica en China y las placas recibidas son las que se ven en la Fig. 4.5, (a) frente y (b) dorso.

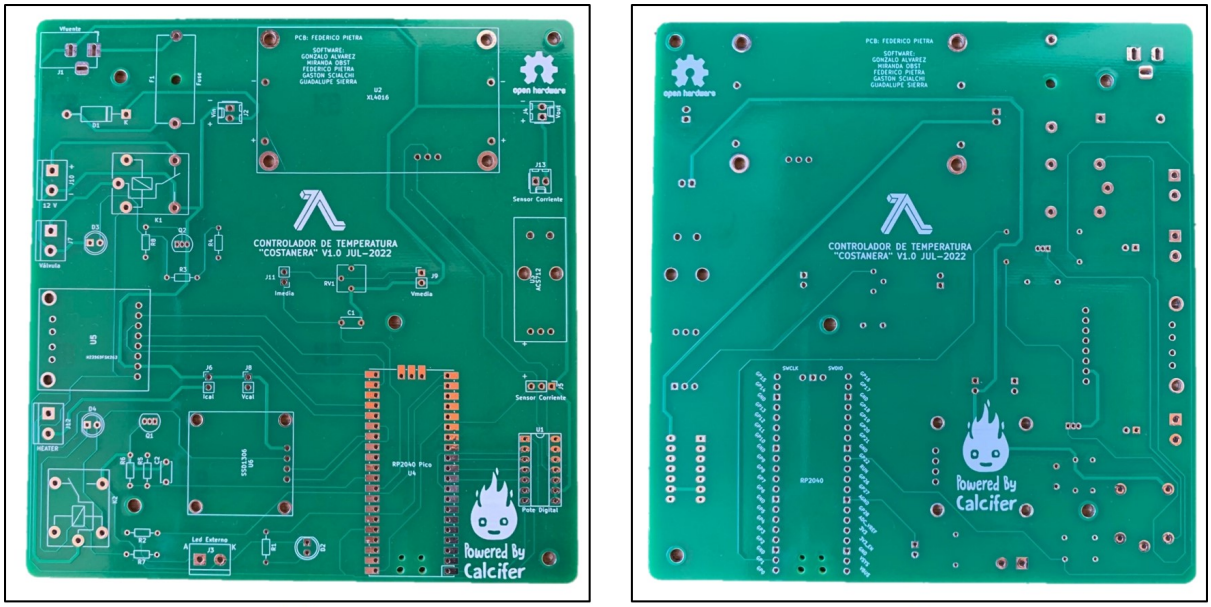

Figura 4.5: Placa PCB fabricada, del sistema completo: (a) frente y (b) dorso.

Finalmente, en la Fig. 4.6 se observa en (a) una imagen 3D en KiCAD y en (b) la placa final ya soldada y ensamblada, con todos sus componentes.

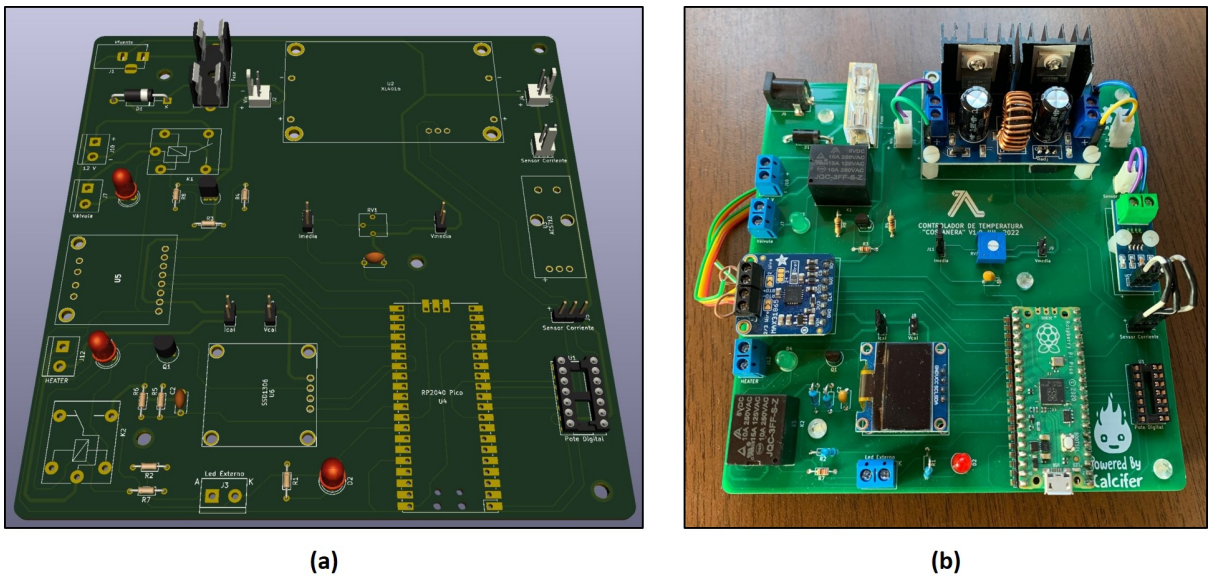

Figura 4.6: (a) Imagen 3D de la placa en KiCAD. (b) Placa PCB completa y terminada, con todos sus componentes incorporados.

La Tabla 4.1 incluye la referencia, descripción y costo aproximado de los componentes a Julio-2022, lo que da un total aproximado de $ 12742 o USD 42,47 (con una tasa de cambio de $300 = USD 1)

Referencia	Valor	Descripción	Precio u.
R1, R2	330 Ω	*Resistencia de 1/4 W, 5 %*	$ 5
R3, R7	220 Ω	*Resistencia de 1/4 W, 5 %*	$ 5
R4, R8	1 KΩ	*Resistencia de 1/4 W, 5 %*	$ 5
R5	10 KΩ	*Resistencia de 1/4 W, 1 %*	$ 5
R6	2 KΩ	*Resistencia de 1/4 W, 1 %*	$ 5
RV1	100 KΩ	*Potenciómetro tipo preset horizontal*	$ 165
C1, C2	100 nF	*Capacitor cerámico multicapa*	$ 22
Q1, Q2	BC547	*Transistor NPN, 50 V/ 100 mA*	$ 25
D1	SR5200	*Diodo Schottky de 5 A*	$ 184
D2	LED	*LED rojo 5 mm*	$ 18
D3, D4	LED	*LED verde 5 mm*	$ 18
F1	5 A	*Fusilera y fusible de 5 A*	$ 100
K1, K2	JQC-3FF-S-Z	*Relé de 5 V/ 10 A*	$ 240
U1	MCP41HV51-502	*Pote. digital de 5 KΩ , 8 bit, HV*	$ 400
U2	XL4016	*Fuente BUCK de 40 V/ 8A regulable*	$ 1600
U3	ACS712-5A	*Sensor de corriente por efecto Hall, 5 A*	$ 940
U4	RP2040 Pico	*Microcontrolador Raspeberry Pi Pico*	$ 1500
U5	Max31865	*Conversor A/D para medición de temp.*	$ 900
U6	SSD1306	*Display OLED de 128x64 píxeles*	$ 960
J1	Jack	Conector hembra 2.1 circ.	$ 90
J2, J4, J13	Molex	*Conector tipo Molex de 2 pines*	$ 25
J3, J7, J10, J12	Bornera	*Bornera con tornillo (2 conectores)*	$ 65
Fuente	19,5 V/ 3,42 A	*Fuente de notebook Toshiba*	$ 4000
Varios	-	Pines, cables, zócalo 20x2, 7x2, PCB	$ 900
TOTAL			**$ 12742**

Tabla 4.1: Lista de componentes y costo.

4.2. Desarrollo del software de control

Para operar los módulos incluídos en la PCB se desarrolló el software de control en un marco de código abierto de forma que cualquier persona pueda utilizarlo y modificarlo. Con estos objetivos se implementó el código dentro del paradigma de **clases** (de python) y se utilizó `git` para el control de versiones haciéndolo disponible en un repositorio: **GitLab**. Cada grupo utilizó su propio repositorio, de manera de trabajar con sus propios códigos a gusto, para luego unificarlos en la versión completa cuyo framework fue desarrollado por uno de los grupos de laboratoristas.

Este esquema de desarrollo dió una solución a la necesidad de llevar un control de versiones de software, de manera tal de permitir obtener una versión estable y funcional, y a la vez poder incorporar modificaciones sólo cuando ya se determinó que funcionan. Nuestro esquema de desarrollo consiste en una rama principal (o *main branch*), que contiene la versión que funciona correctamente, y otras ramas donde se hacen las pruebas/modificaciones. Luego, se pueden incorporar estas ramas a la principal (proceso denominado *merge*) e ir construyendo la versión completa y validada una vez comprobado que no existan conflictos.

Si se utiliza Linux como sistema operativo, normalmente `git` viene preinstalado, y si se usa Windows habrá que instalar un cliente como Git Bash. A modo de referencia, a continuación

se detallan los comandos principales utilizados para el manejo de los códigos en el repositorio:

```
0 git clone git@gitlab.com  # inicializa un repositorio local, clonado del
  ↪ repositorio remoto indicado como url "git@gitlab.com"
1 git status                # permite ver en que branch estoy y se actualizó
2 git add Test.py           # agrega file Test.py al repositorio
3 git commit -a -m "Hola"   # sube el file al repo remoto con comentario "Hola"
4 git push origin main      # asocia el file a la branch main
5 git branch Nueva          # crea una branch "Nueva"
6 git checkout main         # cambia a la branch "main"
7 git merge Nueva           # pasa lo que hay en branch "Nueva" a branch "main"
```

En mi caso, utilicé `https://gitlab.com/koala13/tesis`, donde se pueden encontrar las rutinas principales que desarrollé y que funcionan correctamente, con su correspondiente descripción. Dichas rutinas son las siguientes:

- clase `max31865.py`: Permite controlar un sensor de temperatura RTD conectado a la placa conversora A/D Adafruit Max31965
- clase `intLed.py`: Se utiliza para regular la intensidad de un LED usando el PWM y el ADC de la Raspberry Pi Pico
- clase `gpio.py`: Se utiliza para configurar los puertos de la Raspberry Pi Pico
- `display1306_Test3.py`: Contiene ejemplos de programación de un display OLED 1306
- `mcp41HV51-502_Test4.py`: Ejemplo de manejo de potenciómetro digital MCP41HV51-502 (5 KΩ, 256 pasos). También mide tensión y corriente con ADC_0 y ADC_1 (corriente con sensor de Efecto Hall ACS712-5A)
- clase `fuenteXL4016.py`: Permite regular la tensión de salida de una fuente switching basada en el XL4016, a través de una resistencia variable de 10 KΩ (digital), usando pines de control de la Raspberry Pi Pico
- `Led+Temp+Fuente.py`: Mide temperatura con RTD (import max31865). Regula intensidad de LED con un potenciómetro mecánico externo de 10 KΩ e indica intensidad con un vúmetro (import intLed). Incluye ejemplo de manejo de interrupciones al presionar un pulsador. También regula la tensión de una fuente basada en XL4016 y MCP4131-103 (import fuenteXL4016)
- `calibracion_fuente.py`: Permite calibrar los ADC para medir V o I. Se configura la cantidad de puntos a medir (M) y la cantidad de mediciones por punto (N), y promedia. Devuelve un array con los valores de V e I (digitalizados con los ADC), para cada punto M. Esos valores luego se ingresan manualmente en calibracion_fuente_graficos.py
- `calibracion_fuente_graficos.py`: Con los valores de calibracion_fuente.py y las mediciones con multímetro se ajusta por polinomio de grado definible por usuario y devuelve

los parámetros de ajuste, junto con los gráficos (en "y" lo medido con multímetro y en "x" lo obtenido con ADC). Luego los parámetros de ajuste se ingresaran manualmente en el código principal (main)

- `main_test12.py`: Este código se instala en la Raspberry Pi Pico (bajo el nombre "main.py"). Permite recibir comandos por terminal (con código test12.py). Tiene funciones de medición de temperatura (con Max31865), medición de potencia (con ADC_0 y ADC_1, y sus ajustes configurados manualmente, con calibracion_fuente_graficos.py), setea intensidad y tiempo de encendido de un LED, flashea un LED ("x" mseg), y muestra todo en un display OLED 1306

- `test12.py`: Código que trabaja en conjunto con main_test12.py: permite ingresar comandos por consola (usando un puerto serial), y controlar a la Raspberry Pi Pico sin la necesidad de uso de Thonny

En la Fig. 4.7 se ilustra el menú de uno de los ejemplos de códigos de prueba desarrollado: `test12.py`. Dicha rutina se ejecuta por línea de comando teniendo la Raspberry conectada por USB, para poder comunicarse por el puerto serial sin necesitar Thonny. Entonces para conocer el port de uso, si se ejecuta en Windows se puede averiguar con el comando `C:\> mode` o ejecutando `C:\> wmic path Win32_SerialPort`. Luego se ingresa el número de COM en el código `test12.py`. Si en cambio se utiliza MAC OS o linux, se puede conseguir el puerto serial ejecutando `# ls /dev/*` y tomando nota de `/dev/tty.usbserial****` o `/dev/tty.usbmodem****`.

Como se ve en el menú la Fig. 4.7, la rutina permite: medir temperatura con Max31865, con la opción `-t`, regular la intensidad de un LED con la opción `-l` y flashear un LED con la opción `-f`.

Si se elige medir temperatura, se puede especificar un determinado DELTA de tiempo entre mediciones (opción `-d tiempo_seg`) y la información se salva en un archivo *temperaturas.csv* junto con el valor del RTD y el horario en que se hizo la medición. Si se elige regular un LED, se puede especificar la intensidad entre 0 y 100 % (opción `-i valor_intensidad`) y el período prendido (opción `-p tiempo_seg)`. Finalmente, si se requiere flashear el LED, se puede elegir el tiempo por el cual esta prendido (con la opción `-o tiempo_mseg`). También permite medir la potencia entregada por la fuente a través de la conversión de los ADC, tanto para el caso de la corriente como el de la tensión, previamente habiendo ejecutado el proceso de calibración ya descripto. Toda esa información también es presentada en un display OLED 1306 de 128x64 píxeles a través de diferentes pantallas, junto con el logo de LAMBDA.

```
(base) PS C:\Users\pietra\Desktop> python .\test10.py --h
Conectado al puerto serial COM5...

usage: test10.py [-h] [-t] [-f] [-l] [-d DELTA] [-o ON] [-p PERIODO]
                 [-i INTEN] [--version]

Este programa mide temperaturas con una PT100 y un sensor Adafruit MAX 31865,
regula intensidad y tiempo prendido de un Led, y flashea otro Led.

optional arguments:
  -h, --help  show this help message and exit
  -t          Mide temperatura.
  -f          Flashea el Led.
  -l          Controla el Led (intensidad y tiempo prendido).
  -d DELTA    Medicion temperatura: Tiempo DELTA entre mediciones de
              temperatura, en [seg]. Default: 1 seg.
  -o ON       Flash Led: Tiempo ON del Led, en [mseg]. Default: 1 seg.
  -p PERIODO  Control Led: Duracion PERIODO del Led prendido, en [seg].
              Default: 1 seg.
  -i INTEN    Control Led: Intensidad del Led [0-100]. Default: 50.
  --version   show program's version number and exit
```

Figura 4.7: Menú de opciones para uno de los códigos de prueba desarrollados; permite medir temperatura, controlar intensidad/tiempo de un LED y prender/apagar otro LED.

Por el lado de la Raspberry, se requiere cargar el código `main_test12.py` (que también se encuentra en el repositorio) en la Raspberry con el nombre `main.py` para que se ejecute automáticamente apenas se conecte. Esa rutina tiene 5 funciones principales que dependen de los comandos del puerto serie, provenientes de `test12.py`:

- `readTemp()`: Mide con la RTD conectada a la Max31865 y calcula su temperatura, luego repite cada cierto intervalo de tiempo
- `readPot(level)`: Mide tensión y corriente con ADC_0 y ADC_1 al configurar el potenciómetro digital en "level" (de 0 a 256), luego ajusta con polinomios que se ingresan a mano, provenientes de los códigos de calibración
- `medirADC(Nro_adc, N_veces)`: Recibe el "Nro_adc" (0, 1 o 2) a medir y el número de mediciones "N_veces" a promediar. Devuelve la tensión proporcional en ese ADC
- `setIntensity(intensidad,duracion)`: Controla "intensidad" de un LED, que se prende por "duracion" segundos
- `flashLED(miliseconds)`: Prende un LED durante un tiempo "miliseconds"

Capítulo 5

Medición de desempeño

Este capítulo tiene el objetivo de determinar el impacto en las mediciones con Skipper-CCD, cuando se incorpora el sistema de control desarrollado. Con este fin, por un lado se realizó un procedimiento para determinar el ruido de lectura generado y por otro, se llevó a cabo una prueba en detección de partículas.

5.1. Ruido de lectura y detección de partículas

En principio, en ausencia de partículas que interactúen en el silicio del detector, se espera encontrar todos los píxeles con CERO electrones. Ahora bien, al hacer múltiples mediciones y realizar un histograma, lo que ocurre mas precisamente es que se obtiene una distribución Gaussiana centrada en cero e^- con cierta varianza que se reduce al aumentar el número de muestras.

Cuando no todos los píxeles se encuentran vacíos, algunos de ellos tendrán uno, dos, tres, etc. electrones, y por lo tanto el espectro medido consistirá de Gaussianas centradas en cada uno de esos valores. Estos electrones se pueden generar por distintos motivos, por ejemplo debido a la corriente oscura, y la información que brindan la utilizamos en la calibración.

En la Fig. 5.1 se observa una imagen tomada con Skipper-CCD con tiempo de exposición de 4 horas, donde las trazas blancas corresponden a ionizaciones de diferentes partículas. En particular, se pueden ver trazas producidas por el paso de muones (lineas rectas) y eventos de menor tamaño, producidos por radiación ambiente, como así también eventos de un electrón producidos, por ejemplo, por corriente oscura. Luego, el histograma expone el valor de los píxeles para una región sin trazas, donde se observan picos para pixeles con 0, 1 2 y 3 electrones.

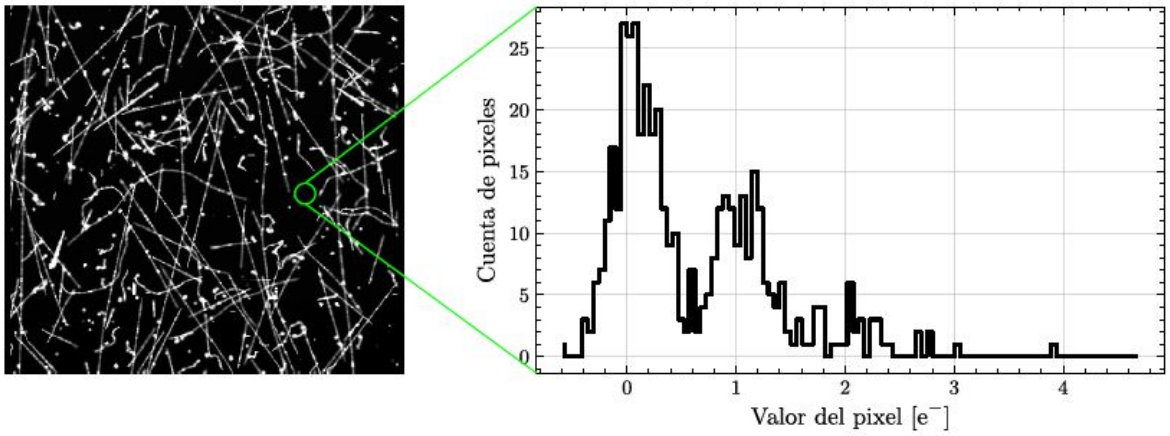

Figura 5.1: (Izquierda) Imagen de partículas detectadas con Skipper-CCD y (derecha) su correspondiente histograma de una región sin trazas, donde se observa mayor concentración de píxeles con cero y un electrón.

Las distribuciones Gaussianas estan definidas por la Ec. 5.1, representada en un ejemplo en la Fig. 5.2, donde N es un factor de normalización, x es la variable aleatoria, μ es el valor medio y σ la desviación estándar. Este último parámetro, equivalente a la raíz cuadrada de la varianza, es el parámetro que determina el ancho de la campana. Luego la incerteza de cada pixel queda determinada por la desviación estándar (σ), y en este contexto se redefine como el **ruido de lectura**.

$$f_{(x)} = Ne^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{5.1}$$

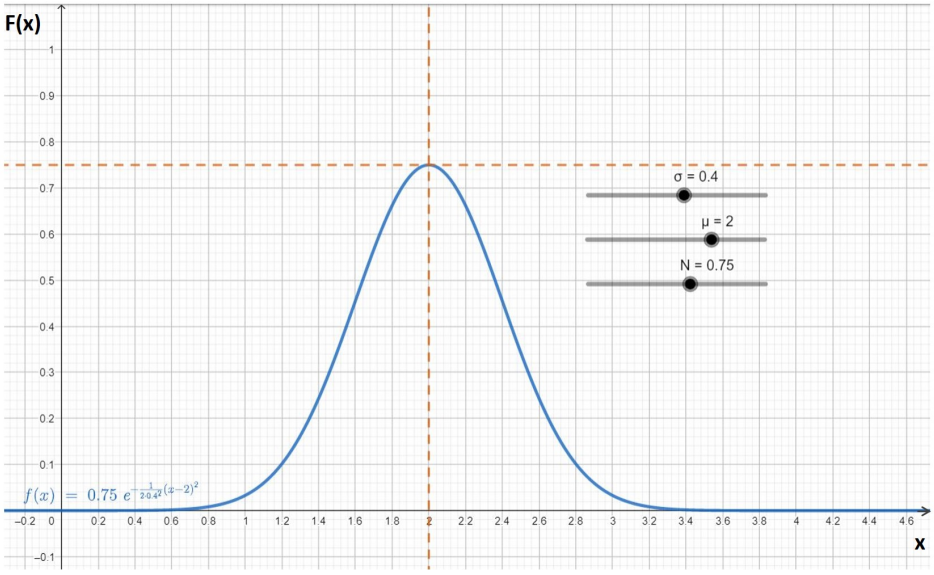

Figura 5.2: Ejemplo de distribución Gaussiana con media $\mu = 2$, factor de normalización $N = 0,75$ y desviación estándar $\sigma = 0,4$.

Para poder evaluar el impacto en el ruido de lectura en un Skipper-CCD al utilizar el sistema de control desarrollado, se enfrió un sistema a aproximadamente 130 K y luego se midió dicho ruido para el caso $N_{SAMPLE} = 1$, y para el caso $N_{SAMPLE} = 25$. Para ambas

situaciones se comenzó midiendo con la fuente de corriente apagada, y luego aumentando la potencia de a 10 % hasta llegar al máximo posible (100 %). En todas las mediciones se computó el valor de temperatura al que se encuentra el sensor. Notar que esa temperatura no es igual a la temperatura del heater debido a la inercia térmica. Al aumentar la potencia que entrega la fuente, aumenta consecuentemente la temperatura del heater hasta que el sistema llega a valores que escapan del rango deseable (entre 130 y 150 K), entonces en esos casos se regula la libración de vapor de nitrógeno de manera tal de bajar la temperatura, por lo tanto se observa que la temperatura de la CCD fluctúa y no aumenta progresivamente conforme aumenta la potencia.

En la Fig. 5.3 se grafica el ruido electrónico en función de la potencia de la fuente. Las barras azules indican las mediciones realizadas solo con $N_{SAMPLE} = 1$ y las naranja con $N_{SAMPLE} = 25$. Además, en color rojo se muestra la temperatura a la cual se realizó cada medición.

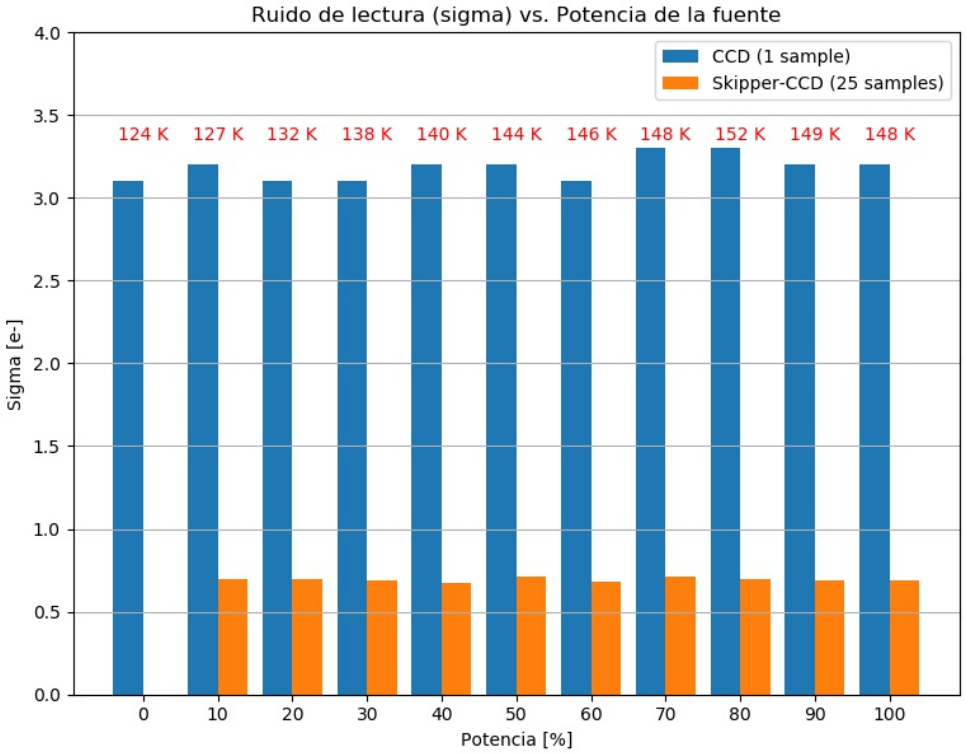

Figura 5.3: Ruido de lectura σ vs. Potencia de la fuente (porcentual), tanto para mediciones con 1 muestra (barras azules) como para 25 muestras (barras naranjas). En color rojo se informa la temperatura de medición.

Es importante destacar que, tanto para el caso de medición con $N_{SAMPLE} = 1$ como con $N_{SAMPLE} = 25$, el ruido de lectura no se ve impactado por la fuente de potencia desarrollada. En las mediciones con $N_{SAMPLE} = 1$, el ruido obtenido es $\sigma_0 = (3{,}2 \pm 0{,}1)e^-$ y para el caso de 25 muestras, el resultado es $\sigma = (0{,}69 \pm 0{,}02)e^-$. Nótese también que ambos valores se relacionan a través de la expresión $\sigma = \sigma_0/\sqrt{N}$, según lo explicado en el capítulo 1. En este caso, $\sigma = (3{,}2 \pm 0{,}1)e^-/\sqrt{25} = (0{,}64 \pm 0{,}02)e^-$, lo que se asemeja a los valores obtenidos en forma práctica.

5.2. Régimen de ultra bajo ruido

Para evaluar el desempeño del sistema de control en el régimen de ultra bajo ruido de lectura, se realizó una medición con 400 muestras utilizando el sistema encendido y funcionando. En la Fig. 5.4 se exhibe el histograma correspondiente, donde se puede observar un pico predominante en CERO electrones y un pico secundario en UN electrón, ambos ajustados por una función Gaussiana, con una desviación estándar o ruido de lectura $\sigma = (0{,}17 \pm 0{,}01)e^-$.

Al igual que los casos descriptos anteriormente, se cumple la relación $\sigma = \sigma_0/\sqrt{N} = (3{,}2 \pm 0{,}1)e^-/\sqrt{400} = (0{,}160 \pm 0{,}005)e^-$. Esto demuestra que utilizar el modo Skipper resulta en mediciones que permiten contabilizar electrones literalmente de a uno, sin ser afectado por el sistema de control.

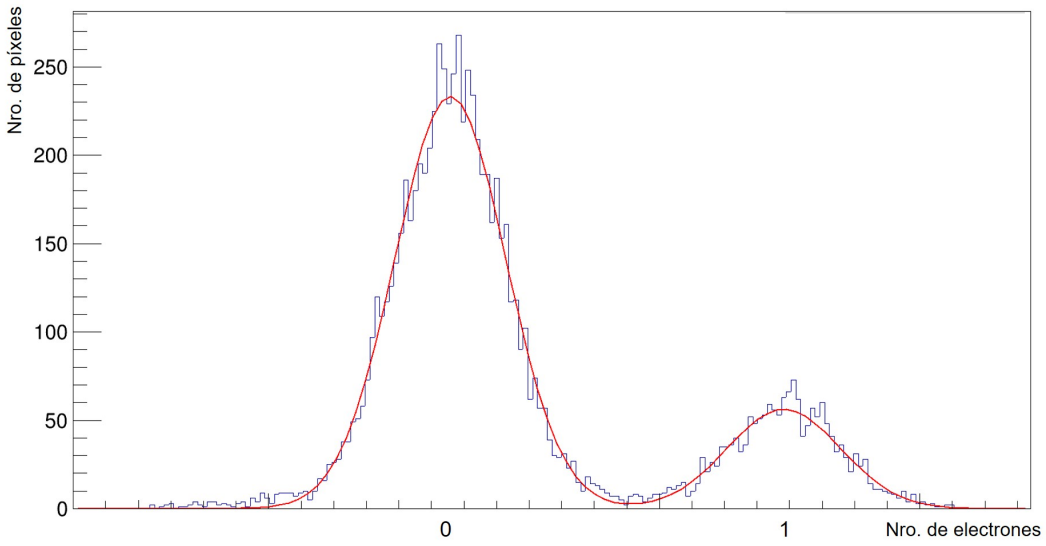

Figura 5.4: Histograma de carga por píxel de un Skipper-CCD, con 400 muestras, ajustado por Gaussiana centrada en 0 y 1 e^- con $\sigma = (0{,}17 \pm 0{,}01)e^-$.

Vale la pena resaltar que no sería posible contabilizar con total precisión la carga de cada uno de estos eventos de no haber alcanzado ruido subelectrónico mientras se operaba la placa desarrollada en el marco de esta tesis.

Capítulo 6

Conclusiones

El trabajo de tesis inició familiarizándome y aprendiendo a utilizar el microcontrolador Raspberry Pi Pico y la plataforma de programación Thonny. En una segunda etapa, aprendí a utilizar la herramienta de desarrollo de esquemáticos y PCB (KiCAD), que me permitió integrar el hardware de manera segura y replicable. Por el lado de la programación, aprendí el concepto de clases de python, al igual que la utilización de un repositorio (GitLAB). Éstas herramientas fueron críticas ya que el proyecto requirió la colaboración entre grupos que realizaron distintas partes con el objetivo de desarrollar un producto de código abierto, compartible con la comunidad.

En paralelo, dado que dichas herramientas se utilizaron para desarrollar un sistema de control para las mediciones con Skipper-CCD, también resultó importante entender el funcionamiento de dicho sensor, el procedimiento de detección de partículas y sus características, sobretodo en lo que refiere al ruido de lectura.

Consecuentemente, el sistema de control desarrollado cuenta con un mecanismo de medición de temperatura que utiliza un conversor A/D y amplificador (Max31865), el cual funciona en conjunto con la resistencia de platino PT-100 ubicada dentro de la cámara de vacío del Skipper. Dicho módulo se utilizó satisfactoriamente en reiteradas mediciones, para monitorear y controlar el enfriamiento del Skipper y luego su temperatura de operación.

Después de un proceso de investigación y testeo elegimos incorporar una fuente de potencia regulable por software, capaz de entregar hasta 19,50 V/ 3,42 A (66,69 W) al heater (que también se encuentra dentro de la cámara de vacío). La regulación de potencia que suministra la fuente se hace con un algoritmo PID (desarrollado por un grupo de laboratorio 6 y 7) que permite converger mas rápidamente a la temperatura deseada. La regulación por software mencionada se basa en la utilización de un potenciómetro digital MCP41HV51-502 de 5 KΩ, el cual permite obtener una salida de potencia aproximadamente lineal, discretizada en 256 pasos. También evalué la posibilidad de utilizar otros potenciómetros digitales que eventualmente pueden resultar de utilidad, aunque el mejor resultado se obtuvo con el circuito elegido. Para complementar la función de la fuente, agregué un mecanismo de encendido/apagado basado en un relé mecánico.

Con el objeto de monitorear la potencia que entrega la fuente, incorporé un módulo de telemetría que utiliza dos conversores A/D del microcontrolador. Éstos puertos son ciertamente imprecisos, por lo que agregué un mecanismo de calibración para comparar las mediciones de

tensión y corriente con un multímetro y luego ajustarlas por un polinomio de grado definible por el usuario, lo que permite obtener telemetría mas confiable.

Además, el sistema de control cuenta con un mecanismo de regulación de intensidad y tiempo de encendido de un LED alocado dentro de la cámara de vacío, que permite calibrar el sensor. La intensidad y tiempo se configura por línea de comando, a través de un puerto serial, y se informa en un display OLED de 128x64 píxeles, junto con otro tipo de información, como ser la potencia entregada por la fuente y la temperatura de operación del Skipper.

Finalmente incorporé un módulo que controla una electroválvula, responsable de liberar o no vapor de nitrógeno que ocasiona que se enfríe al sistema. De esa forma, se puede reducir el consumo de nitrógeno y regular la temperatura a primer orden.

Una vez ensamblado el PCB, lo conectamos a la cámara de vacío del Skipper y realizamos mediciones del ruido de lectura , tanto con 1 muestra, como con 25 muestras. Para ver si la corriente inyectada en el heater por nuestro controlador introduce ruido en el sistema, hicimos un barrido en la potencia de la fuente; inicialmente apagada, hasta llegar al 100 % de su capacidad. Se observó que el ruido es, respectivamente, $\sigma_0 = (3{,}2 \pm 0{,}1)e^-$ y $\sigma = (0{,}69 \pm 0{,}02)e^-$, en toda la escala de potencia. Eso concluye que la utilización del sistema de control no afecta al ruido de lectura, y por lo tanto puede utilizarse durante la toma de datos de ciencia, y también corrobora la relación existente entre mediciones con Skipper y sin Skipper: $\sigma = \sigma_0/\sqrt{N}$. Dicha relación y el impacto de la fuente también se verificó en una medición con 400 muestras, cuyo ruido resultó ser $(0{,}17 \pm 0{,}01)e^-$.

Por último quisiera incluir una foto en LAMBDA del grupo de trabajo conformado para la realización de este proyecto (Fig. 6.1), donde se encuentran los estudiantes Miranda Obst y Guadalupe Sierra, Gastón Scialchi y Gonzalo Alvarez, los directores Javier Tiffenberg, Ana Martina Botti y Darío Rodrigues (lamentablemente Miguel Sofo Haro no pudo estar presente).

Figura 6.1: Foto del grupo de trabajo para el desarrollo del sistema de control. De izquierda a derecha: Gonzalo Alvarez, Gastón Scialchi, Javier Tiffenberg, Federico Pietra, Miranda Obst, Darío Rodrigues, Guadalupe Sierra, Ana Martina Botti.

Bibliografía

[1] A. G. Wright. The photomultiplier handbook. *Oxford University Press*, 2017.

[2] G. Hall. Semiconductor particle tracking detectors. *Reports on progress in physics. Physical Society (Great Britain)*, 57(5):481–531, 1994.

[3] G. Fernandez-Moroni, R. Harnik, P. A. N. Machado, I. Martinez-Soler, Y. F. Perez-Gonzalez, D. Rodrigues, and S. Rosauro-Alcaraz. The physics potential of a reactor neutrino experiment with skipper-ccds: searching for new physics with light mediators. *Journal of high energy physics*, 2022(2), 2022.

[4] J. R. Janesick. Scientific charge-coupled devices. *Press Monographs. Society of Photo Optical*, 2001.

[5] C. C. Hu. Modern semiconductor devices for integrated circuits. *Pearson, 1st edition*, 2009.

[6] A. E. Chavarria, J. I. Collar, J. R. Peña, P. Privitera, A. E. Robinson, B. Scholz, C. Sengul, J. Zhou, J. Estrada, F. Izraelevitch, J. Tiffenberg, J. R. T. de Mello Neto, and D. Torres Machado. Measurement of the ionization produced by sub-kev silicon nuclear recoils in a ccd dark matter detector. *Phys. Rev. D*, 94:082007, Oct 2016.

[7] J. Tiffenberg, M. Sofo-Haro, A. Drlica-Wagner, R. Essig, Y. Guardincerri, S. Holland, T. Volansky, and T-T. Yu. Single-electron and single-photon sensitivity with a silicon skipper ccd. *Physical Review Letters*, 119(13), Sep 2017.

[8] L. Barak, I. M. Bloch, A. Botti, M. Cababie, G. Cancelo, L. Chaplinsky, F. Chierchie, M. Crisler, A. Drlica-Wagner, R. Essig, J. Estrada, E. Etzion, G. Fernandez Moroni, D. Gift, S. E. Holland, S. Munagavalasa, A. Orly, D. Rodrigues, A. Singal, M. Sofo Haro, L. Stefanazzi, J. Tiffenberg, S. Uemura, T. Volansky, and T-T. Yu. Sensei: Characterization of single-electron events using a skipper-ccd. 2021.

[9] J. R. Janesick, T. S. Elliott, A. Dingiziam, R. A. Bredthauer, C. E. Chandler, J. A. Westphal, and J. E. Gunn. New advancements in charge-coupled device technology: sub-electron noise and 4096 x 4096 pixel ccds. *Proc. SPIE 1242, Charge-Coupled Devices and Solid State Optical Sensors*, 1990.

[10] M. Sofo Haro. *Sensores multipixel CCD de ultra bajo ruido de lectura para detección de partículas*. PhD thesis, Instituto Balseiro, Universidad Nacional de Cuyo, 2017.

[11] D. Z. Freedman. Coherent effects of a weak neutral current. *Physical review D: Particles and fields*, 9(5):1389–1392, 1974.

[12] D. Akimov, J. Albert, P. An, C. Awe, P. Barbeau, and B. Becker. Observation of coherent elastic neutrino-nucleus scattering. *Science*, 357(6356):1123–1126, 2017.

[13] A. Aguilar-Arevalo, X. Bertou, C. Bonifazi, G. Cancelo, A. Castañeda, B. Cervantes Vergara, C. Chavez, J. C. D'Olivo, J. C. dos Anjos, J. Estrada, A. R. Fernandes Neto, G. Fernandez Moroni, A. Foguel, R. Ford, J. Gonzalez Cuevas, P. Hernández, S. Hernandez, F. Izraelevitch, A. R. Kavner, B. Kilminster, K. Kuk, H. P. Lima, Ma. Makler, J. Molina, P. Mota, I. Nasteva, E. E. Paolini, C. Romero, Y. Sarkis, M. Sofo Haro, I. M. S. Souza, J. Tiffenberg, S. Wagner, and CONNIE Collaboration. Exploring low-energy neutrino physics with the coherent neutrino nucleus interaction experiment. *Physical review. D. (2016)*, 100(9), 2019.

[14] D. Rodrigues, G. Moroni, C. Bonifaci, and J. C. D'Olivo. Neutrino interaction observation with a low energy threshold array. *Neutrino 2020*, Fermilab, USA, 2022.

[15] F. Zwicky. On the masses of nebulae and of clusters of nebulae. *The Astrophysical Journal*, 86:217, 1937.

[16] Telescope H S. Abell S1063 [J/OL]. https://hubblesite.org/contents/media/images/2018/56/4293-Image.html?page=7&filterUUID=4c394bbb-b21e-43ab-a160-2a4521d70243.

[17] P. A. Ade, N. Aghanim, C. Armitage-Caplan, M. Arnaud, M. Ashdown, and F. Atrio-Barandela. Planck 2013 results. xv. cmb power spectra and likelihood. *Astronomy & Astrophysics*, 571:A15, 2014.

[18] M. Battaglieri. U.s. cosmic visions: New ideas in dark matter. *Essig, J. Mardon, and T. Volansky, Phys. Rev*, 85, 2017.

[19] L. Bergström. Dark matter evidence, particle physics candidates and detection methods. *Annalen der Physik*, 524(9–10):479–496, 2012.

[20] G. Bertone, D. Hooper, and J. Silk. Particle dark matter: evidence, candidates and constraints. *Physics reports*, 405(5–6):279–390, 2005.

[21] F. H. Izraelevitch. *Búsqueda de materia oscura mediante la medición de producción de ionización por retrocesos nucleares con el detector DAMIC*. PhD thesis, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires, 03 2017.

[22] R. Essig, J. Mardon, and T. Volansky. Direct detection of sub-gev dark matter. *Phys. Rev. D*, 85:076007, Apr 2012.

[23] M. Sofo Haro, A. Soto, G. Moroni, F. Chierchie, L. Stefanazzi, R. Chavez, A. Castaneda, K. Hernandez, T. Zmuda, N. Wilser, E. Paolini, A. Oliva, and . Cancelo. A low noise digital readout system for scientific charge coupled devices. In *2017 XVII Workshop on Information Processing and Control (RPIC)*. IEEE, 2017.

[24] Raspberry Pi, RP2040 Raspberry Pi Pico. *An RP2040-based microcontroller board*, 2020.

[25] Maxim Integrated Products,, MAX31865. *Cold-Junction Compensated Thermocouple-to-Digital Converter*, 2011. Rev. 0.

[26] G. C. Goodwin, S. F. Graebe, and M. E. Salgado. Classical pid control. *Pearson*, 2000.

[27] J. W. Webb and R. A. Reis. Pid control and continuous processes. *Prentice Hall*, 1999.

[28] W. K. Roots. Fundamentals of temperature control. *Academic Press, Inc*, 1969.

[29] XLSemi,, XL4016. *8A 180KHz 40V Buck DC to DC Converter*. Rev. 1.2.

[30] Microchip Technology, MCP41HV51-502. *7/8-Bit Single, +36V (±18V) Digital POT with SPI Serial Interface and Volatile Memory*, 2013. Rev. B.

[31] Allegro Microsystems, ACS712-5A. *Hall Effect-Based Linear Current Sensor*.

[32] Solomon Systech, SSD1306. *128 x 64 Dot Matrix OLED/PLED Segment/Common Driver with Controller*, 2009. Rev. 0.4.

[33] OMRON Corporation, CSM-SSR-TG-E-9-2. *Technical Explanation to Solid-state Relays*, 2011.

[34] Texas Instruments,, LM2596. *Power Converter 150-kHz 3-A Step-Down Voltage Regulator*, 1999. Rev. 04.2021.

[35] Microchip Technology, MCP4131-103. *7/8-Bit Single/Dual, SPI Digital POT with Volatile Memory*, 2008.