



UNIVERSIDAD DE QUINTANA ROO
DIVISIÓN DE CIENCIAS E INGENIERÍA

DESARROLLO DE UN SISTEMA ELECTRÓNICO DE
CONTROL E INSTRUMENTACION REMOTA PARA UN
RADIÓMETRO

TESIS
PARA OBTENER EL GRADO DE
MAESTRO EN MECATRÓNICA

PRESENTA
Andrés Omar Kim Batun

DIRECTOR
Dr. Freddy Ignacio Chan Puc

ASESORES
Dr. Guillermo Becerra Nuñez
Dr. Jorge Ovidio Aguilar Aguilar
Dra. Edith Osorio de la Rosa
Dr. Rafael Castrejón García





UNIVERSIDAD DE QUINTANA ROO
DIVISIÓN DE CIENCIAS E INGENIERÍA

TRABAJO DE TESIS BAJO LA SUPERVISIÓN DEL COMITÉ
DEL PROGRAMA DE MAESTRÍA Y APROBADA COMO
REQUISITO PARA OBTENER EL GRADO DE:

MAESTRO EN MECATRÓNICA

COMITÉ DE TESIS

DIRECTOR:

Dr. Freddy Ignacio Chan Puc

ASESOR:

Dr. Guillermo Becerra Nuñez

ASESOR:

Dr. Jorge Ovidio Aguilar Aguilar

ASESOR:

Dra. Edith Osorio de la Rosa



UNIVERSIDAD DE
ASESOR:
QUINTANA ROO
AREA DE TITULACION

Dr. Rafael Castrejón García



CHETUMAL, QUINTANA ROO, MÉXICO, NOVIEMBRE DE 2020.

Resumen

En este proyecto se diseñará e implementará un sistema electrónico para el control e instrumentación de un radiómetro. El radiómetro ha sido patentado en su diseño mecánico y óptico por parte de la Universidad Nacional Autónoma de México (UNAM) y con participación de un investigador de la Universidad de Quintana Roo (UQROO). Sin embargo, está pendiente el diseño de la parte electrónica en la que se realice la instrumentación y control del dispositivo. Por ello se propone en este trabajo de tesis, el desarrollo de un sistema de instrumentación y control del radiómetro.

Se implementa una etapa de adquisición y acondicionamiento de señales para la termopila que funciona como sensor principal. De igual manera, se requiere un sistema de control de temperatura de la termopila para mantener la precisión de la medición. El control desarrollado trabaja mediante un sistema programado PID a una celda Peltier, y se emplea la técnica de Modulación de Ancho de Pulso (por sus siglas en inglés, PWM), para el control de voltaje en dicha celda.

Se implementa un algoritmo de control e instrumentación basado en un sistema embebido, el cual incluye conectividad y una interfaz para la visualización de las mediciones.

Este proyecto de tesis consta de cuatro capítulos en los que se describirá todo lo desarrollado para el sistema de control antes mencionado hasta la conclusión del trabajo. En el primer capítulo se describe la introducción, la justificación, objetivos y alcances del trabajo de tesis, el segundo capítulo está enfocado al funcionamiento del radiómetro óptico lineal para medición de altos flujos radiativos con apuntamiento preciso la selección de los componentes que integran el control para que el funcionamiento del dispositivo sea el óptimo, en el tercer capítulo se presenta en diseño y desarrollo del sistema de control de temperatura donde se habla sobre el hardware que se usó para la construcción del circuito y el software de programación, por último en el cuarto capítulo se muestran los resultados y conclusiones obtenidos.

Tabla de contenido

Resumen.....	1
Índice de Figuras.....	4
Índice de tablas.....	5
Capítulo 1 Introducción.....	6
Objetivo general.....	7
Objetivos específicos.....	7
Planteamiento del problema.....	8
Justificación.....	8
Alcance y aportaciones.....	9
Diseño metodológico.....	9
Capítulo 2 Marco de referencias.....	11
2.1 ¿Qué es un radiómetro?.....	11
2.2 Radiómetro óptico lineal para medición de altos flujos radiativos con apuntamiento preciso.....	12
2.3 Funcionamiento del radiómetro óptico lineal para medición de altos flujos radiativos con apuntamiento preciso.....	13
2.4 Celda Peltier en el radiómetro óptico lineal para medición de altos flujos radiativos con apuntamiento preciso.....	13
2.5 Termopila.....	15
2.6 Celda Peltier.....	16
2.7 ¿Qué es un sistema embebido?.....	17
2.8 Sistema embebido NodeMCU ESP8266.....	18
2.9 Sistema de adquisición de datos y acondicionamiento de señales [11].....	19
2.9.1 Componentes para la instrumentación en un sistema de adquisición de datos.....	20
Capítulo 3 Diseño y Desarrollo.....	25
3.1 Convertidor PWM para acondicionamiento de la señal de la celda Peltier.....	25
3.2 Caracterización de la celda Peltier.....	26
3.3 Control PID digital.....	30
3.3.1 Circuito Eléctrico.....	31
3.4 Sensor de temperatura lm35 y su integración con el sistema embebido NodeMCU ESP8266.....	32
3.4.1 Especificaciones del sensor lm35.....	32
3.4.2 Código de programa.....	33

3.5 Código ID de arduino para el sistema embebido NodeMCU ESP8266.....	34
inicio del Código.	34
Configuración general del código.....	36
Lectura de la variable de entrada del sistema y señal de salida.	37
3.6 Sensor mlx90614 y su integración con el sistema embebido NodeMCU ESP8266.	38
Especificaciones.	38
Código de programa.	39
3.7 Aplicación Blynk.	40
3.7.1 Instalación y creación de la interfaz gráfica en la plataforma BLINK para monitoreo y almacenamiento de datos del sistema de control de temperatura del radiómetro óptico.	41
Código de programa.	47
3.8 Modulo de lectura y escritura de tarjetas microSD [14].	49
3.8.1 Modo de operación del módulo microSD.	50
3.8.2 Características del módulo microSD.	50
3.8.3 Código de programa.....	51
3.9 Diagama de flujo del programa.	53
Capítulo 4 Pruebas experimentales.	54
4.1 Banco de pruebas.	54
4.1.1 Prueba uno: control de temperatura con puente “H” con el sistema embebido Arduino UNO.....	54
4.1.2 prueba dos: control de temperatura con transistor BJT TIP31C a lazo abierto con el sistema embebido Arduino UNO.	55
4.1.3 prueba tres: control de temperatura con transistor BJT TIP31C a lazo cerrado con el sistema embebido Arduino UNO.	56
4.1.4 Prueba cuatro: código del programa en el sistema embebido Arduino Uno.	57
4.2 Migración de componentes electrónicos del sistema embebido Arduino Uno al sistema embebido NodeMCU ESP8266.	59
4.3 Mediciones.	61
4.3.1 Equipo utilizado.....	62
4.3.2 medición de temperatura regulada a 25 °C por medio de control PID.....	62
4.4 Conclusiones	68
4.5 Trabajos futuros.	68
Referencias.	70
Anexos.	72
Código completo.	72

Índice de Figuras.

Figura 2.1. Vista en explosión de componentes ópticos y mecánicos del radiómetro [22].	12
Figura 2.2 Sistema óptico del radiómetro [22].	13
Figura 2.3 Corte transversal del ensamble (radiómetro) [22].	14
Figura 2.4 Esquema de conexión para medición de temperatura de un termopar.	15
Figura 2.5. Conexión Interna de una termopila.	16
Figura 2.6. Componentes de una celda Peltier [6].	17
Figura 2.7 Estructura de puertos de salida del sistema embebido NodeMCU ESP8266 [10].	19
Figura 2.8 Sistema de adquisición de datos[11].	19
Figura 2.9 Proceso de cuantificación.	22
Figura 3.1 Circuito acondicionador de señal PWM para celda Peltier.	25
Figura 3.2 Representacion de celda Peltier [13].	26
Figura 3.3 Comportamiento de la temperatura Vs voltaje con polarización directa.	27
Figura 3.4 Comportamiento de la temperatura Vs corriente con polarización directa.	28
Figura 3.5 Comportamiento de la temperatura Vs voltaje con polarización inversa.	29
Figura 3.6 Comportamiento de la temperatura Vs corriente con polarización inversa.	29
Figura 3.7 Esquema de un sistema de control PID digital a lazo cerrado con acondicionamiento de señal acoplado a la planta [12], [23].	30
Figura 3.8 Diagrama electrónico del sistema de control de temperatura.	31
Figura 3.9 Sensor LM35 [18].	32
Figura 3.10 Esquema de conexión utilizada en el sensor LM35.	32
Figura 3.11 Lectura y conversión de la señal analógica del sensor LM35.	34
Figura 3.12 Configuración inicial del código.	34
Figura 3.13 Configuración general del código	37
Figura 3.14 Lectura de la señal de entrada.	37
Figura 3.15 Diagrama a bloques del sensor MLX90614.	38
Figura 3.16 Configuración inicial del programa del sensor MLX90614.	39
Figura 3.17 Impresión de la lectura de datos del sensor MLX90614.	40
Figura 3.18 Interfaz de la aplicación Blynk [16].	40
Figura 3.19 Sistema de conexión de dispositivos a servidores Blynk.	41
Figura 3.20 Inicio de sesión de la aplicación Blynk.	41
Figura 3.21 Parte uno: Creación de un nuevo proyecto.	42
Figura 3.22 Parte dos: Creación de un nuevo proyecto.	42
Figura 3.23 Texto de envío de token de autenticación.	43
Figura 3.24 Agregar elementos a la interfaz gráfica.	44
Figura 3.25 Elementos agregados a la interfaz gráfica.	45
Figura 3.26 Exportación de datos sensados en archivo CSV.	46
Figura 3.27 Envío del archivo de datos al correo electrónico vinculado con la cuenta.	46
Figura 3.28 Interfaz gráfica creada en la aplicación Blynk.	47
Figura 3.29 Librerías utilizadas en el sistema embebido NodeMCU ESP8266.	47

Figura 3.30 Configuración de conexión a una red WIFI y token de autenticación.....	48
Figura 3.31 Inicio de conexión de red y monitor serial.....	48
Figura 3.32 Configuración de puertos virtuales para enviar datos a la aplicación Blynk....	49
Figura 3.33 Modulo lector de microSD [15].....	49
Figura 3.34 Librerías y variables de inicio para código de módulo de tarjetas microSD. ...	51
Figura 3.35 Inicialización del sensor MLX90614 y modulo microSD.	51
Figura 3.36 Lectura y escritura y creación de archivo para almacenamiento de datos en la microSD.	52
Figura 3.37 Algoritmo del programa para el sistema de control de temperatura.	53
Figura 4.1 Circuito de control de temperatura manual con puente H L298.	55
Figura 4.2 Circuito controlador de temperatura con BJT TIP31C a lazo abierto.	56
Figura 4.3 Circuito controlador de temperatura con BJT TIP31C a lazo Cerrado.	57
Figura 4.4 Librerías y variables de inicio del programa.....	58
Figura 4.5 Cuerpo completo de programa de control de temperatura a lazo cerrado.	59
Figura 4.6 Sistema de control migrado al sistema embebido NodeMCU ESP8266.....	60
Figura 4.7 Prototipo del sistema de control de temperatura.....	61
Figura 4.8 Dispositivos y componentes utilizados para el diseño del sistema de control de temperatura.	61
Figura 4.9 Fuente de alimentación y osciloscopio utilizados.....	61
Figura 4.10 Temperatura ambiente de celda Peltier.	62
Figura 4.11 Medición de señal PWM y corriente del sistema de control de temperatura a temperatura ambiente.....	63
Figura 4.12 Medición del sistema de control de temperatura en el ID de Arduino.	64
Figura 4.13 Temperatura regulada a 25 °C con error de 0.1 °C por debajo de lo deseado.	65
Figura 4.14 Temperatura regulada a 25 °C con error de 0.3 °C por encima de lo deseado.	65
Figura 4.15 Medición de señal PWM y corriente del sistema de control de temperatura a regulada a 25.3 °C.	66
Figura 4.16 Medición de señal PWM y corriente del sistema de control de temperatura a regulada a 24.9 °C.	66
Figura 4.17 Medición del sistema de control de temperatura al alcanzar los 25 °C.....	67

Índice de tablas

Tabla 2.1 Sensores y su señal de medición	20
Tabla 3.1 Caracterización de la celda Peltier con polarización positiva.....	27
Tabla 3.2 Caracterización de la celda Peltier con polarización invertida	28

Capítulo 1 Introducción.

En la industria de generación de energía es de especial interés medir cuantitativamente el flujo de radiación térmica, ya que permite evaluar el comportamiento, eficiencia, robustez y otras propiedades a los que son sometidos las fuentes generadoras de energía como altos hornos, fundidoras, calderas, quemadores, sistemas de soldadura o similares y establecer criterios para el diseño adecuado para optimizar el intercambio de calor [1].

Este trabajo está centrado en el desarrollo de un sistema electrónico de control e instrumentación remota para un radiómetro óptico, este se define como un instrumento de medición del flujo de radiación emitida por cuerpos que comúnmente generan energía por ondas electromagnéticas como hornos, calderas, cámaras de combustión, lámparas, etc. Estos instrumentos emiten una cantidad elevada de energía, la cual es transmitida mediante radiación. El radiómetro óptico con el que se trabajó a lo largo de este proyecto consta de una termopila que funciona como sensor principal de radiación, este sensor convierte la energía a un potencial eléctrico proporcional a la radiación.

Uno de los principales parámetros que afectan a la medición de radiación del radiómetro óptico es la temperatura ambiente del lugar donde se encuentre ubicado el dispositivo, ya que el sensor tiene que estar a una temperatura constante de 25 °C para que la medición de energía sea precisa, por lo que se diseñó y construyó un sistema de control para regular la temperatura de la termopila por medio de la técnica de Modulación por Ancho de Pulso (por sus siglas en inglés PWM) [1].

Para el desarrollo del sistema de control de control e instrumentación remota se utilizó el sistema embebido NodeMCU *ESP8266*, el sistema de instrumentación consiste en el monitoreo de las variables eléctricas de voltaje generado por el sensor de radiación (termopila) y la temperatura del dispositivo, el sistema permite la visualización y almacenamiento de datos en un servidor. Los datos almacenados se pueden descargar en archivos con formato CSV vía WEB desde un servidor FTP instalado en una PC.

El sistema electrónico para el control de temperatura consta de una celda tipo Peltier la cual sirve como sistema de enfriamiento o calefacción del sensor de radiación, esto dependerá del medio ambiente de trabajo donde se encuentre el dispositivo, se controlará la celda para ajustar la temperatura a 25 °C por medio de la técnica PWM que consiste en modificar el ciclo de trabajo del voltaje aplicado a la celda para poder regular la temperatura [2], [3], [4].

Cabe mencionar que la principal aportación de este trabajo es el diseño y construcción de un sistema electrónico de control de la temperatura del dispositivo de medición (termopila) mediante la técnica PWM, así como el diseño de un programa para la adquisición de señales mediante sensores seleccionados específicamente para este trabajo y el almacenamiento de en un servidor FTP mediante un sistema embebido.

Objetivo general.

- Desarrollar e implementar un sistema de control e instrumentación para un radiómetro, que incluya instrumentación local y remota (basado en Internet de las cosas, IoT).

Objetivos específicos.

- Caracterizar una termopila empleada como sensor de radiación en un radiómetro.
- Diseñar e implementar un control de temperatura con técnica PWM para una celda Peltier.
- Diseñar y desarrollar un sistema de adquisición de datos y acondicionamiento de señales para una termopila en un radiómetro.
- Desarrollar una interfaz gráfica para la instrumentación y control remoto.

Planteamiento del problema.

Investigadores de la Universidad Nacional Autónoma de México, con la colaboración de un investigador de la Universidad de Quintana Roo, diseñaron y construyeron un radiómetro óptico lineal para medición de altos flujos radiativos con apuntamiento preciso el cual es capaz de medir la energía radiativa emitida por cuerpos a temperaturas de hasta 6000 °C. El radiómetro requiere de la caracterización del sensor de radiación (termopila) para obtener su respuesta dinámica y eficiencia de conversión. Para ello se requiere medir el voltaje generado por la termopila y la temperatura para obtener su respuesta dinámica en las mediciones del lugar y condiciones donde se encuentre [1].

Justificación.

En este trabajo se propone desarrollar un sistema de control y programación flexible que permita la adquisición de variables eléctricas de voltaje, temperatura mediante la programación del sistema embebido *ESP8266*. Este sistema consta de un circuito de control de temperatura, el cual está basado en una celda tipo Peltier junto con un sensor de temperatura que permitirá el control por medio de la técnica PWM que regulará la temperatura de una termopila. La celda Peltier tiene un soporte conductor térmico en el que se ubica la termopila y un disipador térmico que enfría o calienta a la termopila según sea requerido, para mantenerlo a una temperatura constante de 25 °C independientemente de la temperatura ambiente del lugar donde se encuentre, y así poder aumentar la precisión y exactitud, asegurando la linealidad del radiómetro óptico lineal para medición de altos flujos radiativos con apuntamiento preciso [1] [21].

El sistema igual permite visualizar y almacenar los datos adquiridos. Esto quiere decir que las variables de medición de temperatura y radiación adquiridas se podrán almacenar y descargar en archivos con formato CSV vía WEB desde un servidor FTP (File Transfer Protocol, por sus siglas en inglés) instalado en una PC, los datos almacenados igual se podrán visualizar a través de una aplicación para dispositivos móviles con sistema operativo Android.

Alcance y aportaciones.

El proyecto en desarrollo tiene como alcances los siguientes puntos:

- La caracterización del sensor de radiación de tal manera que se pueda instrumentar las mediciones del radiómetro.
- Diseño e implementación de un acondicionador de señales analógicas para la termopila infrarroja.
- Desarrollar un sistema de adquisición de señales con un sistema embebido.
- Diseñar e implementar un sistema de control de temperatura basada en la técnica PWM para una celda de Peltier con un área de 1cm x 1cm.

Los aspectos puntuales que comprende la investigación estarán referidos a IoT y la programación de un sistema embebido, estos dos aspectos están ligados ya que la programación del sistema embebido comprende el concepto de IoT que se refiere a conectar un dispositivo a internet, para el control del sistema, al igual que visualizar y almacenar los datos de medición.

Los aspectos relacionados con el diseño y construcción del radiómetro, cálculos de medición teórica de radiación, no serán contemplados en este proyecto. No se incluye el costo beneficio de la implementación debido que no se cuenta con parámetros de comparación.

Este proyecto tiene como principal aportación, el desarrollo e implementación de un sistema de control e instrumentación para un radiómetro, que incluya instrumentación analógica y remota (basado en Internet de las cosas, IoT). También se tiene como aportación consolidar las colaboraciones entre instituciones en materia del diseño de prototipos mecatrónicos.

Diseño metodológico

El desarrollo del sistema de control se basa en procesar las señales de potencial eléctrico, que puedan ser utilizadas como información para el monitoreo y

desempeño un radiómetro óptico, el método de trabajo propuesto a seguir se describe a continuación.

- La búsqueda bibliográfica sobre los temas de interés como: el efecto Peltier, control por técnica PWM y programación de sistemas embebidos son de gran importancia para estar a la vanguardia con el tema de este proyecto, por lo que se realizaron hasta el término de la tesis.
- El estudio, diseño y construcción del circuito de control y sensor de temperatura que se implementara será uno de los primeros puntos a abarcar en el desarrollo de este trabajo, ya que será el componente principal del sistema que proporcionará las pautas para la toma de decisiones a la hora de realizar el procesamiento de las señales adquiridas, teniendo el circuito se integrará al prototipo del radiómetro para que sea ubicado en su interior.
- Una vez estudiado y ubicado el sensor construido, se iniciará con el desarrollo y programación del sistema de adquisición de datos en un sistema embebido para que puedan ser procesada las señales, seguido se identificarán y obtendrán sus características, estas servirán para visualizarlas en una aplicación para dispositivos móviles y se almacenaran en un servidor FTP para manipulación de los datos, por último se harán pruebas de medición de radiación y temperatura, al igual se harán pruebas del control de temperatura.

Capítulo 2 Marco de referencias.

2.1 ¿Qué es un radiómetro?

Los radiómetros son dispositivos que miden la cantidad de flujo radiante de la energía electromagnética dentro en un rango específico de longitud de onda o rango de detección, la unidad de medida del flujo radiante se expresa en Watt por metro cuadrado ($Watt/m^2$) la cual es la unidad de medida del flujo radiativo.

Los radiómetros generalmente se usan para detectar y medir la cantidad de energía fuera del espectro de luz visible y se usan para medir luz ultravioleta (UV) o detectar la temperatura al absorber las ondas infrarrojas (IR) que son emitidas de una superficie.

Un ejemplo de uso típico de un radiómetro que mide UV es en la iluminación de los museos, donde la presencia de UV puede ser muy problemática ya que la energía UV acelera el proceso de deterioro de las piezas de los museos debido a su mayor contenido de energía, por lo que cualquier energía por debajo de $400nm$ debe ser filtrada o eliminada [20].

Otro ejemplo de aplicación para un radiómetro es la detección y medición de infrarrojos o IR. Estos se utilizan para detectar y medir el calor en una superficie.

Los radiómetros infrarrojos pueden ser utilizados para detectar sobrecalentamiento en motores y repararlos de manera segura o detección de cortocircuitos en el cableado.

Una característica de los radiómetros es su velocidad de medición esto se debe a que son dispositivos simples que usan un solo sensor con un filtro diseñado específicamente para medir solo en el rango de longitud de onda para el que fue diseñado.

Como se mencionó con anterioridad, en esta tesis se trabaja con un radiómetro infrarrojo denominado “radiómetro óptico lineal para medición de altos flujos radiativos con apuntamiento preciso” este radiómetro usa un sensor llamado

termopila el cual es un dispositivo que convierte la energía térmica de una superficie en energía eléctrica y se habla de este dispositivo detalladamente más adelante.

2.2 Radiómetro óptico lineal para medición de altos flujos radiativos con apuntamiento preciso.

El radiómetro óptico unidireccional de amplio espectro de radiación y de lectura directa es un instrumento de medición capaz de detectar parte de la energía ultravioleta (UV) visible y el infrarrojo hasta la longitud de onda de $30\ \mu\text{m}$ aproximadamente, esta es la principal característica de este radiómetro en comparación con instrumentos similares ya que puede captar todo el espectro de radiación de los cuerpos a alta temperatura, lo cual le confiere la dicha característica de linealidad. Al igual puede medir con exactitud y en forma precisa, el flujo radiativo emitido por superficies de materiales o flamas a altas temperaturas que van desde los $25\ ^\circ\text{C}$ hasta los $6000\ ^\circ\text{C}$. La Figura 2.1 muestra una vista en explosión que detalla todas las piezas ópticas y mecánicas del radiómetro [1].

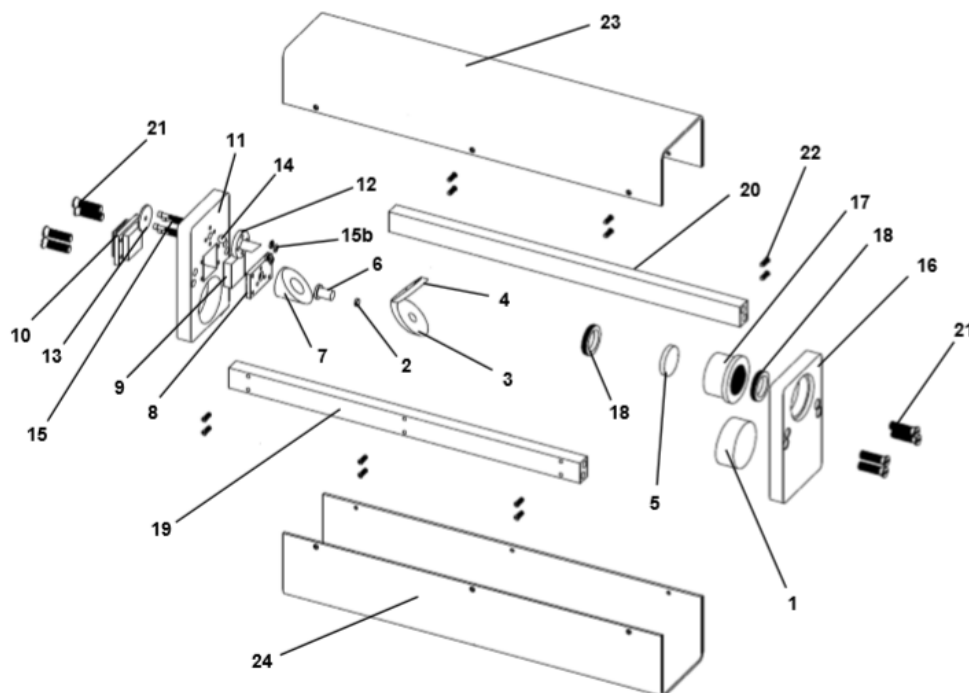


Figura 2.1. Vista en explosión de componentes ópticos y mecánicos del radiómetro [22].

2.3 Funcionamiento del radiómetro óptico lineal para medición de altos flujos radiativos con apuntamiento preciso.

El flujo radiativo entra al radiómetro por una abertura circular que se encuentra en la placa frontal del dispositivo e incide en el espejo parabólico el cual la enfoca al detector termopila. Esta termopila convierte la radiación en un voltaje inducido o fuerza electromotriz (F. E. M.) que es directamente proporcional a la potencia del flujo radiativo que es de interés medir. Parte de la radiación que entra al radiómetro es reflejada por un espejo diagonal con orificio y enviada a un espejo diagonal superior que permite al usuario apuntar a través del ocular a la región la cual es de interés medir. En la Figura 2.2 se muestra un esquema del sistema óptico del radiómetro, donde se puede observar la disposición de los distintos componentes que lo constituyen [1].

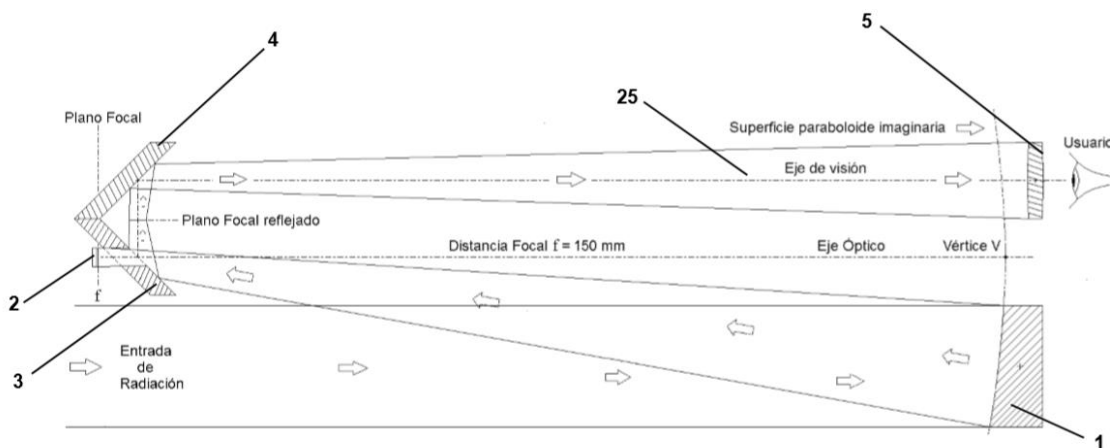


Figura 2.2 Sistema óptico del radiómetro [22].

2.4 Celda Peltier en el radiómetro óptico lineal para medición de altos flujos radiativos con apuntamiento preciso.

Haciendo referencia a la Figura 2.3, el radiómetro óptico lineal para medición de altos flujos radiativos con apuntamiento preciso, tiene un módulo termoeléctrico Peltier que mediante un soporte conductor térmico en el que asienta el detector termopila y un disipador térmico de temperatura este enfría o calienta al detector termopila según sea requerido, para mantenerlo a una temperatura constante de 25 °C independientemente de la temperatura ambiente. Para evitar flujos de calor no

provenientes del detector termopila, un aislador de temperatura térmica está colocado alrededor del soporte conductor térmico, permitiendo solamente el flujo de calor entre el detector termopila y el módulo termoeléctrico Peltier, a través del soporte conductor. El soporte conductor, el aislador y el módulo termoeléctrico Peltier están sostenidos por el disipador que a su vez está alojado y atornillado a la placa frontal. De esta forma, el soporte conductor, el aislador y el módulo termoeléctrico Peltier no hacen contacto conductivo térmico con la placa frontal, evitando así las pérdidas de calor.

La estabilización en la temperatura del detector termopila, mediante la disposición del soporte conductor, el módulo termoeléctrico Peltier y el disipador aumenta la precisión y exactitud, asegurando la linealidad del radiómetro óptico lineal para medición de altos flujos radiativos con apuntamiento preciso.

La Figura 2.3 muestra una vista de corte transversal que muestra el ensamblaje de la parte óptica y mecánica del radiómetro óptico lineal para medición de altos flujos radiativos con apuntamiento preciso [1].

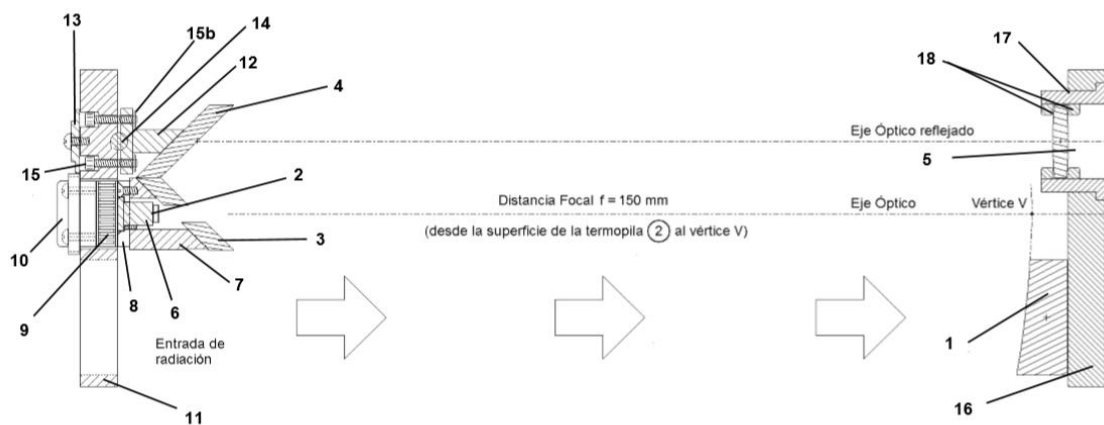


Figura 2.3 Corte transversal del ensamblaje (radiómetro) [22].

2.5 Termopila.

El físico de origen alemán, Thomas Johan Seebeck. Realizó notables investigaciones en varios campos de la física, intentando establecer la conexión entre calor y electricidad. Llegó así a descubrir el efecto Seebeck, en 1821, que consiste en la unión de una lámina de cobre con otra de bismuto (materiales tipo P y N), en un circuito cerrado, que al calentar a la unión de los dos materiales se genera una corriente eléctrica que fluye por el circuito, en tanto se mantenga la diferencia de temperatura. Teniendo como resultado un dispositivo (termopar) que permiten realizar mediciones de temperatura con una gran sensibilidad y precisión. En la Figura 2.4 se muestra un esquema de conexión de un termopar para medir la temperatura de la flama de una vela, el cual está siendo calentada la unión de los materiales por la misma [5], [6], [7].

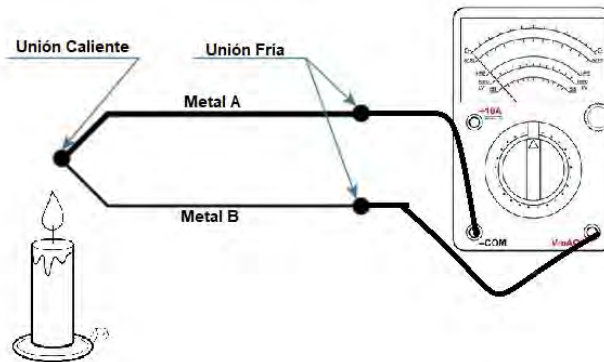


Figura 2.4 Esquema de conexión para medición de temperatura de un termopar
Las termopilas, son dispositivos electrónicos de medición que convierten la energía térmica en energía eléctrica, consisten en varios termopares conectados en serie en todas las uniones frías (ánodo y cátodo) y las uniones calientes (punta de unión de materiales), que se expone a la temperatura que va a ser medida. En la Figura 2.5 se muestra la conexión interna de termopares en serie en una termopila [8].

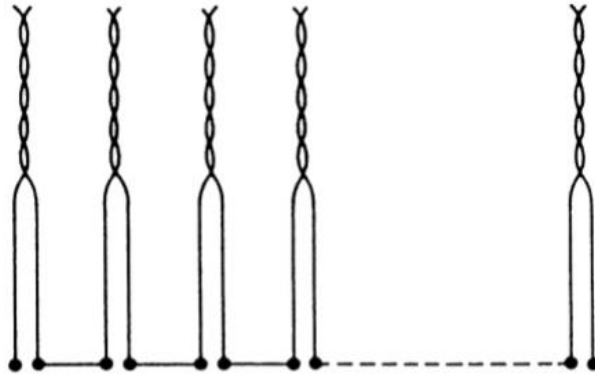


Figura 2.5. Conexión Interna de una termopila.

Otra función de una termopila es la detección de temperatura sin contacto, es decir, obtener mediciones de temperatura a través de un dispositivo que con apuntar a una fuente que emita radiación térmica, este permita la captura de temperatura hacia una termopila y transferir la radiación térmica emitida por el objeto a una salida de voltaje. Otra función de la termopila es determinar temperaturas relativamente bajas (en comparación con el rango de medición de un solo termopar). Algunas aplicaciones donde se usan estos sensores incluyen aparatos como hornos de microondas, secadoras de ropa, dispositivos médicos y automoción (control de clima del automóvil, calefacción de asientos de vehículos) [4].

El efecto de conectar, "n" termopares en serie es aumentar la sensibilidad de medición en un factor de "n" decimales. Un ejemplo de una termopila típica fabricada mediante la conexión de 25 termopares con materiales de cromo y constantán se obtiene una resolución de medición de 0.001 °C [8].

2.6 Celda Peltier.

En 1834, el físico francés, Jean Charles A. Peltier descubrió el "efecto Peltier". El dispositivo que permite realizar este efecto es denominado como Celda Peltier o Dispositivo Termoeléctrico Peltier (TEC) el cual se utiliza principalmente en los sistemas de refrigeración y consiste en la unión de dos metales semiconductores tipo P y N. Cuando el lado del material N se aplica la polaridad positiva de alimentación y en el lado del material P la polaridad negativa, la placa de cobre de la parte superior enfría, mientras que la inferior calienta, si se invierte la polaridad de alimentación, es decir, se aplica en el lado del material N la polaridad negativa y

en el lado del material P la positiva, se invierte la función de calor en la parte superior de la placa mientras que en la parte inferior enfría. En la Figura 2.5 se muestran los componentes que conforman una celda Peltier [5], [6], [7], [9].

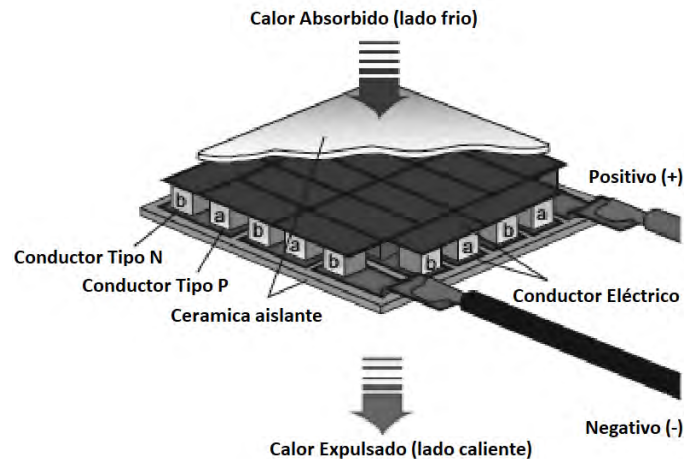


Figura 2.6. Componentes de una celda Peltier [6].

2.7 ¿Qué es un sistema embebido?

Es un dispositivo computacional el cual está diseñado para realizar una o varias tareas en específico, comúnmente se encuentran en dispositivos computacionales en tiempo real. Estos dispositivos, aunque se asemejan a los sistemas computacionales de propósito general (PC), que, si bien también están formadas por una combinación de hardware y software, las computadoras no son diseñadas para un uso específico como son los sistemas embebidos. Una de las principales diferencias que hay entre los sistemas computacionales convencionales y los sistemas embebidos, es que éstos últimos están conectados frecuentemente a ambientes físicos a través de sensores y actuadores [24].

Los sistemas embebidos se construyen generalmente con las siguientes características:

- Eficiencia en cuanto al consumo de energía, peso y costo.
- deben reaccionar a cambios en el ambiente.
- Trabajar en tiempo real, realizar cálculos en tiempo real sin ningún retraso, es decir, en tiempos fijos ante cualquier eventualidad.

- Funcionamiento específico.
- respuestas a una solución óptima de tareas a resolver.
- Son sistemas híbridos, poseen partes analógicas y digitales.
- Capaz de Realizar tareas relativamente complejas.
- Estar constituido por módulos estrictamente indispensables para desarrollar la función encomendada.
- dedicados para ciertas aplicaciones.
- Interfaces de usuario dedicadas (sin ratón, teclado y pantalla).

2.8 Sistema embebido NodeMCU ESP8266

Una de las mejores maneras para el desarrollo rápido de aplicaciones para IoT es el sistema embebido "NodeMCU *ESP8266*". Algunos sistemas embebidos de bajo costo como como son la tarjeta BeagleBone y Arduino siempre han sido una excelente opción para incorporar la automatización en un proyecto. Pero estos módulos tienen un pequeño inconveniente, ya que no cuentan con un módulo WiFi incorporado, por lo tanto, es necesario agregar un módulo WiFi externo en estos dispositivos para que pueda establecer conexión a Internet.

La empresa Espressif Systems diseño el NodeMCU *ESP8266*, que es un sistema embebido de desarrollo electrónico y firmware de código abierto que desempeña un papel importante en el diseño de un producto IoT utilizando unas pocas líneas de script (código). El módulo se basa principalmente en *ESP8266*, el cual es un microchip Wi-Fi de bajo costo que incorpora la capacidad de microcontrolador y combina algunas características del sistema embebido Arduino, pero con la posibilidad de conectarse a Internet por medio del microchip *ESP8266*. En la Figura 2.7 se muestra la estructura de puertos de salida del sistema embebido NodeMCU *ESP8266* [10].

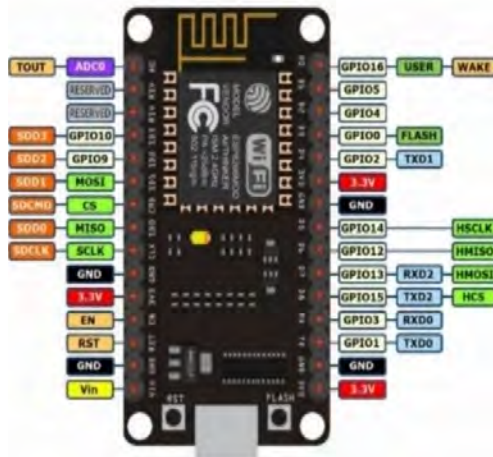


Figura 2.7 Estructura de puertos de salida del sistema embebido NodeMCU ESP8266 [10].

2.9 Sistema de adquisición de datos y acondicionamiento de señales [11].

Por su parte la adquisición de datos (DAQ, Data Acquisition, por sus siglas en inglés) es el proceso de medir con una computadora o sistema embebido por medio de un sensor una señal eléctrica física como voltaje, corriente, temperatura, presión, sonido etc. Un sistema DAQ está formado por sensores, circuitos electrónicos acondicionadores de señal y software para programar el instrumento de medición en un sistema embebido. En la Figura 2.8 se muestra el esquema de un sistema de adquisición de datos basado en PC.



Figura 2.8 Sistema de adquisición de datos[11].

2.9.1 Componentes para la instrumentación en un sistema de adquisición de datos.

Para la implementación de la instrumentación local o remota se requiere de una tarjeta de adquisición de datos la cual contenga circuitos electrónicos acondicionadores de señales, software y una PC con características suficientes para poder ejecutar el software de la tarjeta de adquisición de datos. Para poder realizar la adquisición de datos y obtener buenos resultados en la medición se pueden considerar los siguientes elementos los cuales son esenciales para poder implementar la instrumentación local o remota.

- Sensor o Transductor
- Hardware de acondicionamiento de señales
- PC y Software

Sensores (Transductores).

Los sensores, también llamados transductores, convierten una señal física en una señal eléctrica que se puede medir. Dependiendo del tipo de sensor, la señal eléctrica adquirida por el sensor puede ser una de voltaje, corriente, resistencia u otro variable eléctrica que cambia con el tiempo. Algunos sensores pueden requerir circuitos acondicionadores para producir correctamente una señal que puede ser leída con precisión y con toda seguridad por el hardware de adquisición de datos. En la Tabla 2.1 se muestran algunos tipos de sensores y la señal de medición que producen.

Tabla 2.1 Sensores y su señal de medición.

Sensor	Señal de Medición
Termopar, RTD, Termistor	Temperatura
Fotosensor	Luz
Micrófono	Sonido
Galga Extensiométrica, Transductor Piezoeléctrico	Fuerza y Presión

Potenciómetro, LVDT, Codificador Óptico	Posición y Desplazamiento
Acelerómetro	Aceleración
Electrodo pH	pH

Hardware de adquisición de datos.

El hardware de adquisición de datos actúa como la interfaz entre una PC y las señales analógicas adquiridas por los sensores. Funciona principalmente como un dispositivo que digitaliza señales analógicas entrantes para que una PC pueda interpretarlas. Los tres componentes clave de un dispositivo DAQ usado para medir una señal son: el circuito de acondicionamiento de señales, convertidor analógico-digital (ADC) y un bus de comunicación en la computadora.

Circuitos Acondicionadores de señales.

La señal de salida del sensor de un sistema de medición en general se debe procesar de una forma adecuada para la siguiente etapa de la operación. La señal puede ser, por ejemplo, demasiado pequeña, y sería necesario amplificarla; podría contener interferencias que eliminar; ser analógica y requerir su digitalización; ser digital y convertirla en analógica, etc., a todos estos procesos que se llevan a cabo se le conoce como acondicionamiento de señales. Por ejemplo, las señales de un termopar proporcionan niveles muy pequeños de voltaje que deben ser amplificados antes de la digitalización. Otros sensores, tales como detectores de temperatura mediante resistencias (por sus siglas en inglés RTD), termistores, requieren que se les aplique una fuente de voltaje o corriente para poder funcionar.

Convertidor analógico-digital.

Las señales analógicas de los sensores deben ser convertidas en digitales antes de ser manipuladas por el equipo digital como una PC. Un convertidor analógico-digital (ADC, por sus siglas en inglés) es un chip que proporciona una representación digital de una señal analógica en un instante de tiempo. Las señales analógicas varían continuamente con el tiempo y un ADC realiza "muestras" periódicas de la

señal a una razón predefinida. Estas muestras son transferidas a una PC a través de un bus, donde la señal original es reconstruida desde las muestras en software

Para lograr la conversión de una señal analógica a una señal digital se tiene que hacer una serie de pasos para poder llevarse a cabo dicha conversión los cuales son:

Muestreo.

El muestreo (en inglés, sampling) consiste en tomar muestras periódicas de la amplitud de onda. La velocidad con que se toman esta muestra, es decir, el número de muestras por segundo es lo que se conoce como frecuencia de muestreo (Sampling Frequency) y está en función del teorema de Nyquist, que indica que la frecuencia de muestreo será al menos el doble de la frecuencia de la señal a muestrear.

Cuantificación.

La cuantificación lo que hace es convertir una sucesión de muestras de amplitud continua en una sucesión de valores discretos (digitales) preestablecidos según el código utilizado. Durante el proceso de cuantificación se mide el nivel de voltaje de cada una de las muestras, obtenidas en el proceso de muestreo, y se le atribuye a un valor finito (discreto) de amplitud, seleccionado por aproximación dentro de un margen de niveles previamente fijado. En la Figura 2.9 se muestra el proceso de cuantificación. de una señal analógica.

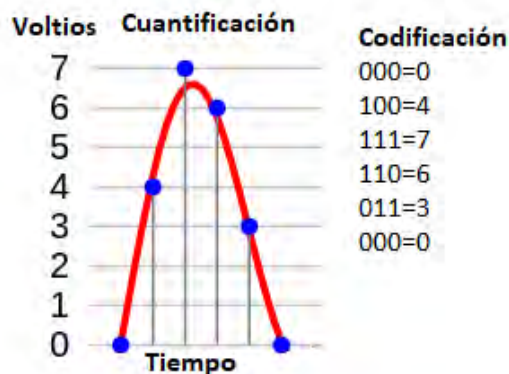


Figura 2.9 Proceso de cuantificación.

Codificación.

La codificación consiste en la traducción de los valores de tensión eléctrica analógicos que ya han sido cuantificados (ponderados) al sistema binario, mediante códigos preestablecidos. La señal analógica va a quedar transformada en un tren de pulsos digital.

Bus de PC.

El hardware de adquisición de datos se conecta a una PC a través de una ranura o puerto. El bus de la PC sirve como la interfaz de comunicación entre el hardware de adquisición de datos y la PC para pasar instrucciones y datos medidos. Los buses de PC más comunes son USB, PCI, PCI Express y Ethernet. Los dispositivos DAQ igual están disponibles para 802.11 Wi-Fi para comunicación inalámbrica.

Función de la PC en un sistema DAQ.

Una PC se usa como dispositivo intermedio entre un software programable y un dispositivo DAQ ya que el software programable es el encargado de controlar las operaciones del dispositivo DAQ y es usado para procesar, visualizar y almacenar datos de medida.

Software de aplicación.

El software de aplicación facilita la interacción entre la PC y el usuario para adquirir, analizar y presentar datos de medidas. Las aplicaciones creadas en el software pueden ser diseñadas con funcionalidades predefinidas o diseñadas con funcionalidades personalizadas. Las aplicaciones personalizadas generalmente son usadas para automatizar múltiples funciones de un dispositivo DAQ, realizar algoritmos de procesamiento de señales y mostrar interfaces de usuario personalizadas.

Software controlador.

El software controlador ofrece al software de aplicación la habilidad de interactuar con un dispositivo DAQ. El software simplifica la comunicación con el dispositivo DAQ al abstraer comandos de hardware de bajo nivel y programación a nivel de

registro. Generalmente, el software controlador DAQ expone una interfaz de programación de aplicaciones (API) que es usada en un entorno de programación para construir software de aplicación.

Capítulo 3 Diseño y Desarrollo.

3.1 Convertidor PWM para acondicionamiento de la señal de la celda Peltier.

Los microcontroladores proporcionan muy poca corriente en sus puertos de salida. Por ejemplo, las salidas digitales del microcontrolador (Atmega328p) del Arduino Uno proporcionan una corriente máxima de 20mA.

La corriente máxima dada por los puertos de salida del sistema embebido Arduino Uno no es suficiente para activar cargas que necesiten una corriente superior, por lo que se necesita algún tipo de circuito de amplificación de corriente y así poder controlar la activación de las cargas desde el microcontrolador.

Al igual que la corriente, el voltaje de los puertos salidas también está limitada y como ejemplo los puertos de salida del microcontrolador Arduino Uno solo proporcionan un voltaje que va de 0V a 5V. Por lo que para activar cargas que necesiten un voltaje superior es necesario un circuito de amplificación. Para activar de forma sencilla cargas desde un microcontrolador, ya sea para proporcionar más corriente o más voltaje, se puede utilizar los transistores BJT en estado de conmutación. La Figura 3.1 muestra el circuito utilizado para activar una celda Peltier a través de una señal PWM proveniente del sistema embebido NodeMCU ESP 8266.

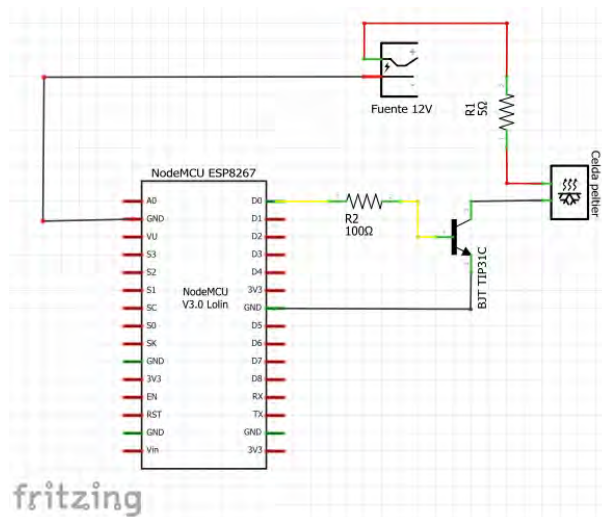


Figura 3.1 Circuito acondicionador de señal PWM para celda Peltier.

En el convertidor PWM para el acondicionamiento de la señal se utilizó un transistor BJT TIP31C este transistor puede soportar un voltaje máximo de 100V y una corriente máxima de 3A, en la Figura 3.1 se puede observar que se colocó una resistencia shunt con un valor de 5Ω en serie con la celda Peltier y la fuente de alimentación de 12V, esto se realizó para que el consumo de corriente de la celda Peltier no exceda la corriente máxima del transistor y así evitar que se dañen los componentes.

3.2 Caracterización de la celda Peltier.

En la Figura 3.2 se muestra representada una celda peltier, Haciendo referencia al capítulo anterior con respecto a la celda peltier, cuando el lado del material N se aplica la polaridad positiva de alimentación y en el lado del material P la polaridad negativa, la placa de cobre de la parte superior emite frío, mientras que la placa de cobre inferior emite calor, con fines prácticos y para una mejor comprensión a este tipo de conexión se le llamara **(polarización positiva o directa)** y si se invierte la polaridad de alimentación, es decir, se aplica en el lado del material N la polaridad negativa y en el lado del material P la positiva, se invierte el funcionamiento de la celda ahora emitiendo calor en la parte superior de la placa mientras que en la parte inferior enfría, al invertir el tipo de conexión se llamara **(polarización negativa o inversa)**.



Figura 3.2 Representacion de celda Peltier [13].

En las siguientes tablas y figuras se muestran los valores de caracterización de una celda Peltier con polarización positiva y polarización invertida conectada directamente a una fuente de alimentación variable, con base a los datos y graficas

mostradas en la hoja de datos de la celda Peltier se puede apreciar que la caracterización realizada es correcta y el comportamiento es el esperado [13].

En la Tabla 3.1 se muestran los valores de voltaje de alimentación a la celda (con base a la hoja de especificaciones), corriente y temperatura en grados Celsius los cuales corresponden a la caracterización de la celda peltier con una polarización positiva.

Tabla 3.1 Caracterización de la celda Peltier con polarización positiva.

Polarización directa		
Voltaje	Corriente	°C
0	0	30
0.2	0.2	27
0.4	0.36	24
0.6	0.5	21
0.8	0.68	19
1	0.8	18.5

La Figura 3.3 muestra el comportamiento de la temperatura de la celda Peltier al aplicarle un voltaje variable con polarización directa.

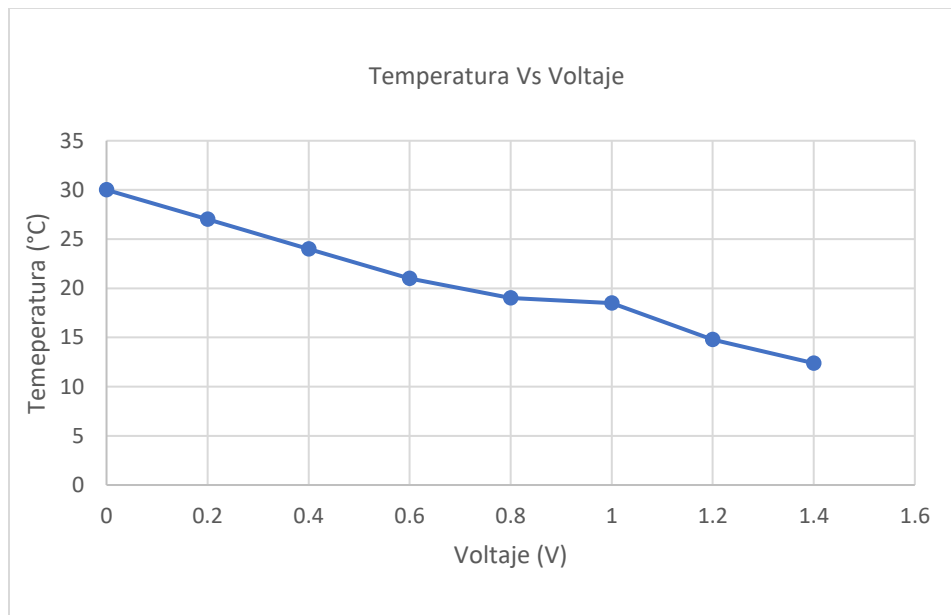


Figura 3.3 Comportamiento de la temperatura Vs voltaje con polarización directa.

La Figura 3.4 muestra el comportamiento de la celda Peltier con polarización directa con respecto al consumo de corriente.

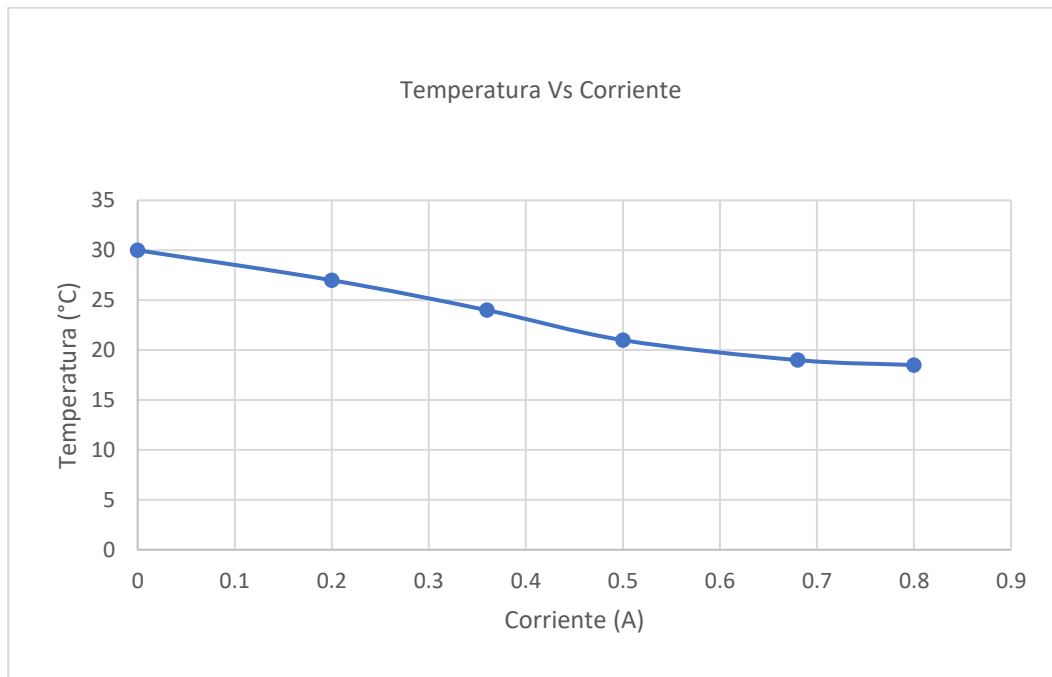


Figura 3.4 Comportamiento de la temperatura Vs corriente con polarización directa.

En la Tabla 3.2 se muestran los valores de voltaje, corriente y temperatura en grados Celsius los cuales corresponden a la caracterización de la celda Peltier con polarización invertida.

Tabla 3.2 Caracterización de la celda Peltier con polarización inversa.

Polarización Inversa		
Voltaje	Corriente	°C
0	0	30
0.2	0.2	31
0.4	0.36	32
0.6	0.5	34
0.8	0.68	37.5
1	0.8	42
1.2	0.9	45.5
1.4	1.02	49
1.5	1.15	52

La Figura 3.5 muestra el comportamiento de la temperatura de la celda de la Peltier al aplicarle un voltaje variable con polarización inversa.

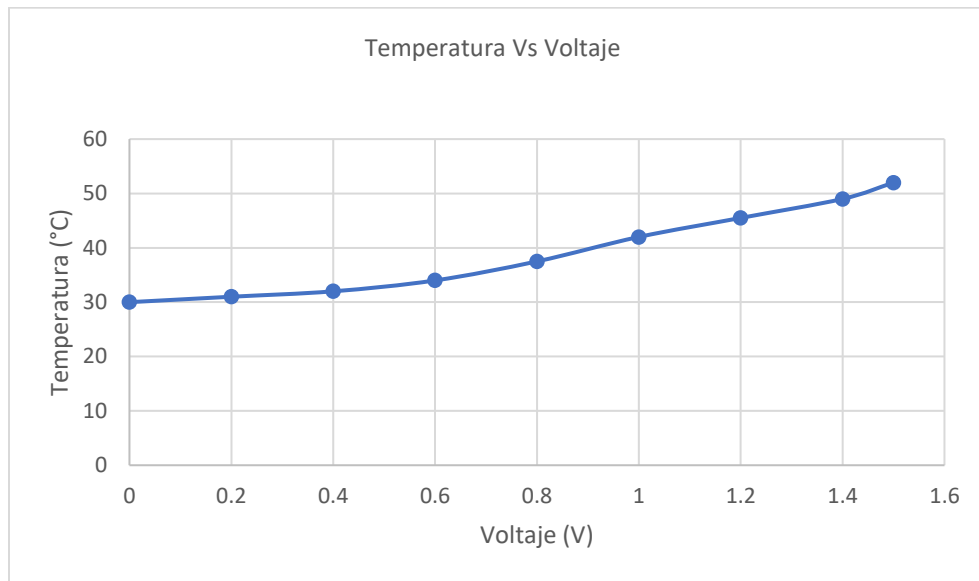


Figura 3.5 Comportamiento de la temperatura Vs voltaje con polarización inversa.

La Figura 3.6 muestra el comportamiento de la temperatura de la celda peltier con polarización invertida con respecto al consumo de corriente.

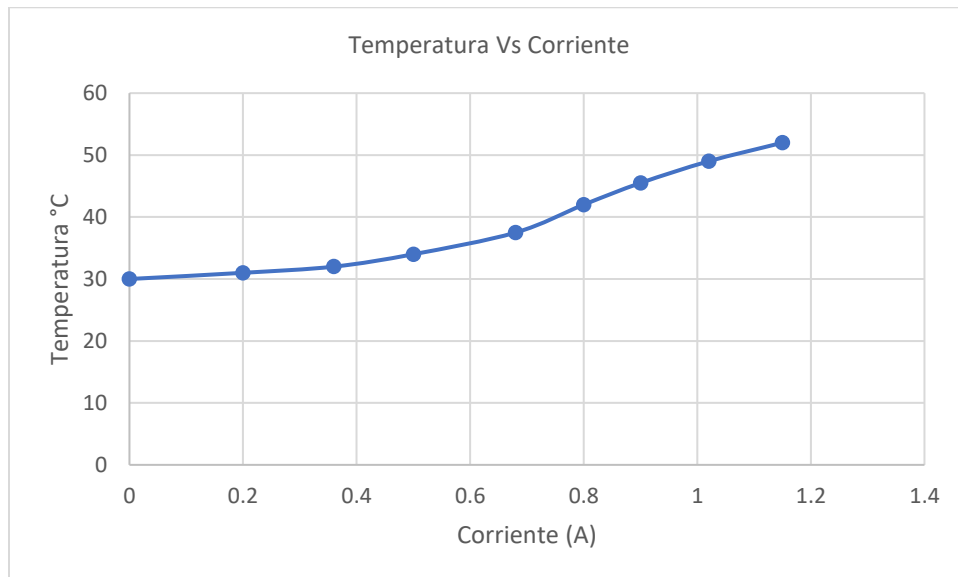


Figura 3.6 Comportamiento de la temperatura Vs corriente con polarización inversa.

3.3 Control PID digital.

El control PID, es un proceso controlado por un sistema de compensación en lazo cerrado, que consta de tres parámetros reguladores de acciones proporcional, integral y derivativa o PID, de tal forma que logra que el sistema responda a cambios en sus variables en una forma estable.

El propósito de un sistema de control digital (o discreto) es reemplazar al dispositivo controlador por un programa digital. Esto se puede lograr con microcontroladores, DSP, FPGAs, etc., teniendo como proceso lo siguiente: La señal de salida del proceso de control digital se muestrea cada cierto intervalo de tiempo (a esto se le llama período de muestreo) y es discretizada o muestreada mediante un convertidor analógico-digital (ADC). Esta información es procesada por el controlador digital y convertida nuevamente en analógica mediante un convertidor digital-analógico (DAC). Por lo tanto, internamente el controlador digital se independiza del tipo de señal con que está trabajando y ve todas las señales como una serie de valores discretos. En la Figura 3.7 se muestra el esquema típico de un sistema de control digital en lazo cerrado [12], [19].

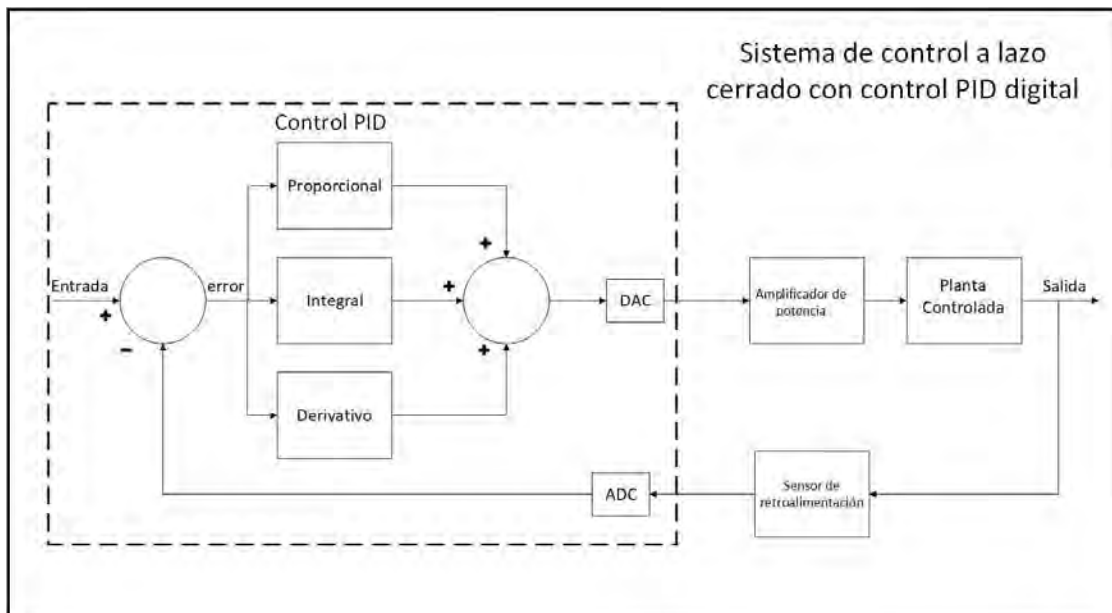


Figura 3.7 Esquema de un sistema de control PID digital a lazo cerrado con acondicionamiento de señal acoplado a la planta [12], [23].

3.3.1 Circuito Eléctrico.

En la Figura 3.8 se puede observar el diagrama electrónico que se utilizó para realizar el control de temperatura para la celda Peltier por medio de un controlador PID digital a lazo cerrado. En esta figura se muestran los componentes que se necesitan para un sistema de control digital a lazo cerrado como se puede observar se encuentra el sistema embebido NodeMCU ESP8266, el cual va a ser el dispositivo que va a proporcionar los valores de control PID para estabilizar el sistema. El convertidor digital-analógico y viceversa el convertidor analógico-digital, al igual se puede observar una resistencia de 100Ω , un transistor BJT TIP31C y una fuente de alimentación de 12V que representa la etapa de amplificación de potencia, seguido puede ver la planta controladora que en este caso es la celda peltier y por último el sensor de retroalimentación el cual se encuentra del lado izquierdo de la figura y está representado por un sensor de temperatura LM35, todos los componentes descritos conforman un sistema de control digital a lazo cerrado como se mostró en la Figura 3.7.

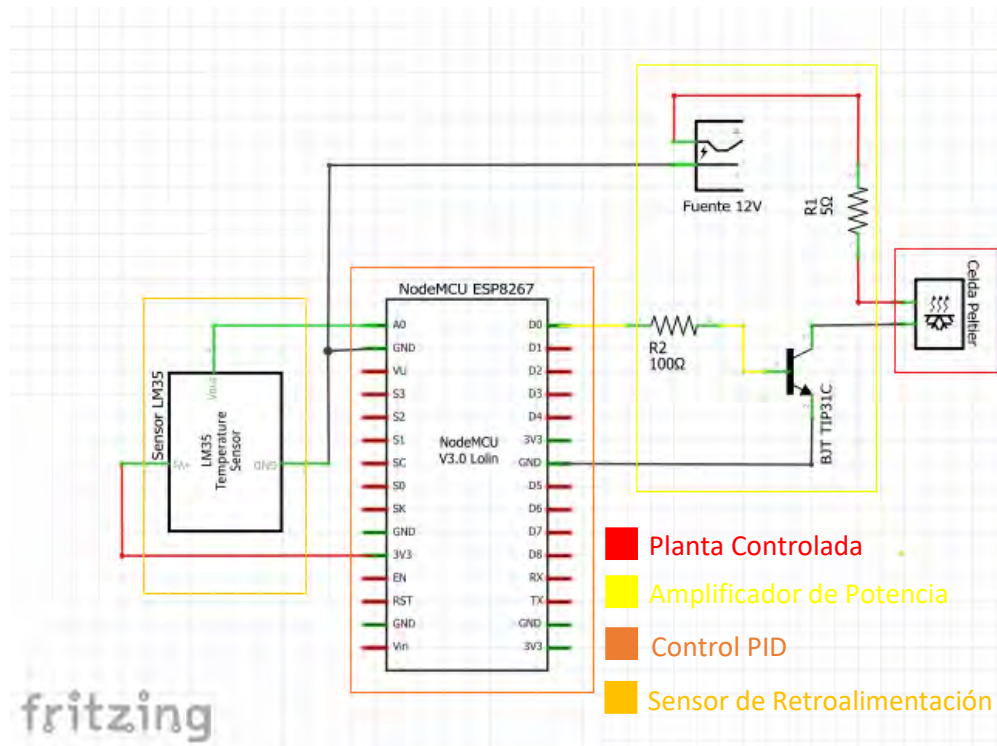


Figura 3.8 Diagrama electrónico del sistema de control de temperatura.

3.4 Sensor de temperatura lm35 y su integración con el sistema embebido NodeMCU ESP8266.

Como ya se ha mencionado anteriormente se utilizó el sensor de temperatura LM35 como señal de entrada del sistema de control.

La señal de temperatura que proporciona el sensor lm35 es la más importante para el diseño del controlador ya que la señal sirve como entrada del control PID y la que se enviará a un puerto de salida para controlar la temperatura de la celda Peltier.

3.4.1 Especificaciones del sensor lm35.

El sensor analógico LM35 de Texas Instruments, nos permite realizar medidas de temperatura de una forma precisa a través de las entradas analógicas de un microcontrolador.

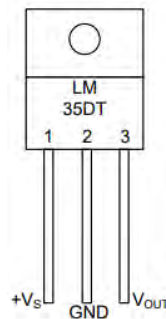


Figura 3.9 Sensor LM35 [18].

Como se puede observar en la Figura 3.9 el sensor consta de 3 pines (VCC, GND y Data), por ello su conexión es sencilla. En la Figura 3.10 se puede observar el esquema utilizado para la conexión al sistema embebido NodeMCU ESP8266.

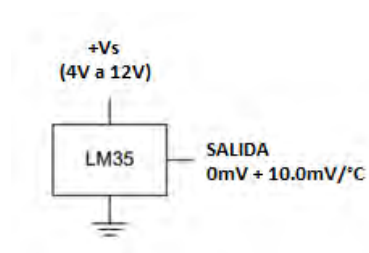


Figura 3.10 Esquema de conexión utilizada en el sensor LM35.

El sensor LM35 presenta las siguientes características:

- Está calibrado directamente en grados Celsius.
- Rango de medición de -55°C a 150°C.
- La tensión de salida es proporcional a la temperatura. Esto quiere decir que 1°C equivale a 10mV.
- Presenta una precisión garantizada de 0.5°C a 25°C.
- Presenta un rango de alimentación entre 4 y 30V.
- Presenta baja impedancia de salida.
- Presenta baja corriente de alimentación (60µA).
- No necesita de circuitos adicionales para su medición o calibración.
- Presenta un coste bastante reducido.

3.4.2 Código de programa.

En la primera parte del código se muestra las variables que se va a calcular y el puerto de entrada de la señal de temperatura del sensor.

La señal del sensor se coloca dentro de un ciclo de treinta iteraciones seguido se divide entre el mismo número de iteraciones para sacar el promedio de las muestras adquiridas en el bucle, una vez obtenido el promedio de las muestras la señal se guardada en la variable Input y la señal analógica se convierte a grados Celsius con la siguiente formula.

$$Input = Input * 100 * 3.3 / 1023 \quad (1)$$

Por último, la señal analógica se imprime en el monitor serial para visualización de los datos censados. En la Figura 3.11 se muestra el código descrito anteriormente.

```
#define PIN_INPUT A0

double Input;

void setup() {
}

void loop()
{
    for (int A = 0; A <= 29; A++) {
        Input = Input + analogRead(PIN_INPUT);
    }

    Input = Input / 30;

    Input = Input * 100 * 3.3 / 1023;

    delay(1000);

    Serial.print("Input : ");
    Serial.print(Input);
}
}
```

Figura 3.11 Lectura y conversión de la señal analógica del sensor LM35.

3.5 Código ID de arduino para el sistema embebido NodeMCU ESP8266.

inicio del Código.

Se inicia el programa agregando las librerías que se utilizaron para el control digital PID y manipulación del sistema embebido NodeMCU *ESP8266*, seguido se definen las variables a calcular y los puertos de salida, como se muestra en la Figura 3.12.

```
control_temp $
#include <ESP8266WiFi.h>
#include <PID_v1.h>

#define PIN_INPUT A0

#define PIN_OUTPUT 16

double Setpoint, Input, Output;

double Kp = 30, Ki = 55, Kd = 0;

PID myPID(&Input, &Output, &Setpoint, Kp, Ki, Kd, REVERSE);
```

Figura 3.12 Configuración inicial del código.

Como se puede observar en la última línea de código que se muestra en la Figura 3.12, está crea un objeto con las variables:

- **Setpoint:** La cual es el valor objetivo que se quiere obtener por medio del controlador. En este caso será la temperatura que queremos que nos proporcione la celda Peltier.
- **Input:** Es la variable que se va a controlar. Será la temperatura para calcular por el control en cada iteración del bucle de control.
- **Output:** Es la variable que nos proporciona la salida de este control. Es decir, el valor obtenido por el sensor de temperatura LM35.

Además de las tres variables mencionadas anteriormente, se tiene tres parámetros declarados dentro del objeto creado los cuales son [19], [23]:

- **Acción de control proporcional (K_p):** esta acción de control es proporcional a la señal de error del sistema que internamente la acción proporcional multiplica la señal del error por la constante K_p esta acción de control intenta minimizar el error del sistema. Cuando la variable del error es grande, la acción de control igual lo es y tiende a minimizar el error, sin embargo, solo con esta acción, el error teóricamente nunca llegará a cero.
Algunas de las características que tiene esta acción al tratar de regular la variable K_p son:
 1. Aumenta la velocidad de respuesta del sistema.
 2. Disminuye el error del sistema.
 3. Si se incrementa mucho la constante K_p de la acción proporcional, aumenta la inestabilidad del sistema.
- **Acción de control Integral (K_I):** esta acción de control se refiere al tiempo que se toma para llevar a cabo la acción correctiva. Mientras el valor sea más pequeño, el ajuste es más rápido. Además, esta acción teóricamente hace que el error tienda a cero

Algunas de las características que tiene esta acción al tratar de regular la variable K_I son:

1. Disminuye el erro del sistema en régimen permanente
 2. Aumenta un poco la velocidad del sistema.
 3. Aumenta la inestabilidad del sistema (si esta acción no es necesaria no se agregar).
- **Acción de control Derivativo (K_D):** esta acción de control emite una acción predictiva, es decir, prevé el error e inicia una acción oportuna. Responde a la velocidad del cambio de la acción de control K_p y produce una corrección significativa antes de que la variable del error se vuelva demasiado grande. (Modifica la respuesta transitoria, por ejemplo, el tiempo de asentamiento, pudiera incrementar el ruido de la medición).

Algunas de las características que tiene esta acción al tratar de regular la variable K_D son:

1. Aumenta la estabilidad del sistema controlado
2. La variable del error permanecerá igual.
3. Disminuye o aumenta la velocidad del sistema

Por último se declara la dirección en la que va a trabajar el control PID el cual está configurado en modo reversa como se muestra en la Figura 3.12, esto quiere decir que como tenemos una entrada mayor a la Setpoint el sistema tiene que enfriar la celda Peltier hasta alcanzar el valor deseado (Setpoint configurado) por lo que en lugar de ir incrementando la entrada del sistema se reduce para alcanzar su objetivo, por eso es que se configura en modo reversa la dirección del control.

Configuración general del código.

En la configuración general del código se declara el modo con el que va a realizar los cálculos de la variable de salida el control PID y la variable Setpoint la cual es la temperatura de salida a la que se quiere regular la celda Peltier como se muestra en la Figura 3.13.

```

void setup()
{
    myPID.SetMode(AUTOMATIC);
    Setpoint = 25;
}

```

Figura 3.13 Configuración general del código

Lectura de la variable de entrada del sistema y señal de salida.

En esta sección del código la lectura de la señal de entrada, que en este caso es la temperatura del sensor LM35 se promedia cada treinta muestras, esto se realiza para obtener una señal estable al momento de graficar la señal.

La función `<<myPID.Compute()>>` que se puede observar en la Figura 3.14 es la que contiene el algoritmo para calcular la salida PID esta cada que se repite el ciclo calcula un nuevo valor para la salida del controlador.

Por último, se llama a la función `<<analogWrite(PIN_OUTPUT, Output)>>` la cual escribe la salida calculada por la función `<<myPID.Compute()>>` y la manda a un puerto de salida de señal PWM con un retraso de medio segundo. En la Figura 3.14 se muestra las líneas de código principal del sistema para el cálculo de las variables de salida PID del sistema.

```

void loop()
{
    for (int A = 0; A <= 29; A++) { //promediador de cada 30 muestras del sensor LM35
        Input = Input + analogRead(PIN_INPUT);
    }

    Input = Input / 30; // Input es entrada de la temperatura (Promediada)

    Input = Input * 100 * 3.3 / 1023; // Pasar Input a °C

    myPID.Compute(); // Calcular el Output

    analogWrite(PIN_OUTPUT, Output); // Escribir el PWM

    delay(500); // Retardo
}

```

Figura 3.14 Lectura de la señal de entrada.

3.6 Sensor mlx90614 y su integración con el sistema embebido NodeMCU ESP8266.

Para simular el funcionamiento del radiómetro óptico lineal para medición de altos flujos radiativos con apuntamiento preciso y así poder hacer pruebas del funcionamiento del sistema de control de temperatura se integró el sensor MLX90614, el cual es un sensor infrarrojo de temperatura sin contacto por medio de la termopila MLX81101, el sensor está diseñado para ser sensible a la radiación infrarroja emitida por un objeto a distancia, este sensor puede obtener lecturas desde los $-70\text{ }^{\circ}\text{C}$ hasta $380\text{ }^{\circ}\text{C}$.

El sensor MLX90614 incluye un amplificador de bajo ruido, un convertor ADC de 17 bits, un DSP (procesador digital de señal) y compensación de la temperatura ambiente. En la Figura 3.15 se muestra el diagrama a bloques del sensor MLX90614 [4].

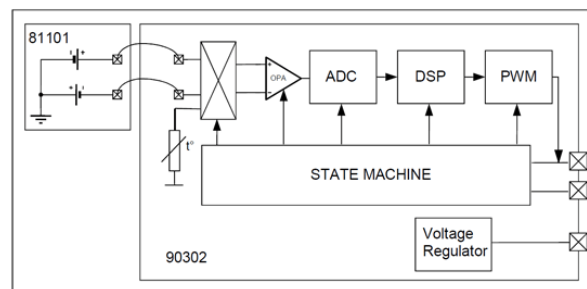


Figura 3.15 Diagrama a bloques del sensor MLX90614.

Especificaciones.

- Voltaje de Operación: 3.3V - 5V
- Rango de temperatura ambiente de trabajo: -40°C hasta $+170^{\circ}\text{C}$
- Rango de temperatura de objeto: -70°C hasta $+380^{\circ}\text{C}$
- Campo de visión: 70° .
- Precisión: $\pm 0.5\text{ }^{\circ}\text{C}$.
- Resolución de medición: $0.02\text{ }^{\circ}\text{C}$
- ADC incorporado de 17 bits
- Salida de PWM: 10 bits
- Protocolo de comunicación SMBUS (I2C)

Código de programa.

En la primera parte del código, para la lectura de datos con el sensor MLX90614 se incluyen las librerías con las que va a estar funcionando el programa, en este caso son tres librerías.

La librería “ESP8266WiFi.h” permitirá establecer conexión con el sistema embebido NodeMCU *ESP8266* y así poder programar la tarjeta.

La librería “Wire.h” permitirá establecer comunicación serial I²C con el sensor para recibir los datos de temperatura procesados.

La librería “Adafruit_MLX90614.h” permitirá establecer la comunicación con el sensor MLX90614 y poder generar los datos de temperatura del sensor.

Seguido se crea un objeto que contendrá la información de configuración preprogramada del sensor.

En la configuración general se inicializa el monitor serial para observar los datos censados, al igual se inicializa la configuración del sensor MLX90614, como se muestra en la Figura 3.16.

```
mlktest$  
#include <ESP8266WiFi.h>  
#include <Wire.h>  
#include <Adafruit_MLX90614.h>  
  
Adafruit_MLX90614 mlx = Adafruit_MLX90614();  
  
void setup() {  
  Serial.begin(9600);  
  
  mlx.begin();  
}
```

Figura 3.16 Configuración inicial del programa del sensor MLX90614.

En esta parte del código como se muestra en la Figura 3.17 se imprime la variable para lectura de radiación infrarroja de objetos y poder visualizarlos en el monitor serial, no se hace utiliza ni una fórmula de conversión de señales gracias a su DSP (procesador de señales digitales, por sus siglas en inglés) interno que trae este sensor.

```

void loop() {
  Serial.print("Temperatura de objeto = "); Serial.print(mlx.readObjectTempC());

  Serial.println();
  delay(500);
}

```

Figura 3.17 Impresión de la lectura de datos del sensor MLX90614.

3.7 Aplicación Blynk.

Blynk es una plataforma de internet de las cosas para teléfonos inteligentes con sistema operativo Android o IOS, esta aplicación para telefonía móvil permite interactuar con sistemas electrónicos desarrollados en las plataformas de Arduino y Raspberry Pi y NodeMCU ESP8266. En la Figura 3.18 se puede observar la interfaz gráfica que maneja la aplicación Blynk.



Figura 3.18 Interfaz de la aplicación Blynk [16].

Con esta herramienta se puede diseñar una interfaz gráfica para el controlar, monitorear y graficar datos obtenidos de sensores conectados a circuitos electrónicos desarrollados en los sistemas embebidos de las plataformas mencionadas anteriormente, al igual esta aplicación es capaz de almacenar los datos obtenidos de sensores. En la Figura 3.19 se muestra el esquema de conectividad que usa la plataforma Blynk [17].

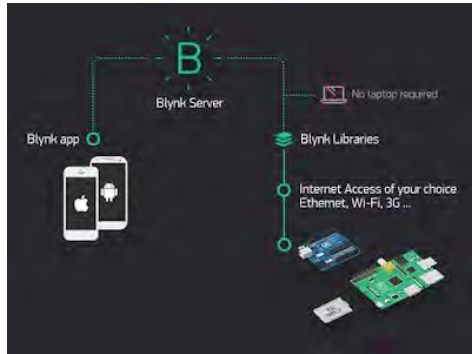


Figura 3.19 Sistema de conexión de dispositivos a servidores Blynk.

3.7.1 Instalación y creación de la interfaz gráfica en la plataforma BLINK para monitoreo y almacenamiento de datos del sistema de control de temperatura del radiómetro óptico.

En esta parte del proyecto se utilizó un teléfono celular con sistema operativo Android, el primer paso que se tiene que hacer es la instalación de la aplicación Blynk esto se puede hacer ingresando a la tienda Google Play buscar la aplicación, descargar e instalar, Una vez instalada la aplicación se ejecuta y se crea una cuenta en la plataforma de Blynk, después de haberse creado la cuenta e iniciado sesión en la aplicación esta ya está lista para crea un nuevo proyecto. En la Figura 3.20 se observa una captura de pantalla que muestra la interfaz de inicio de sesión en la aplicación.

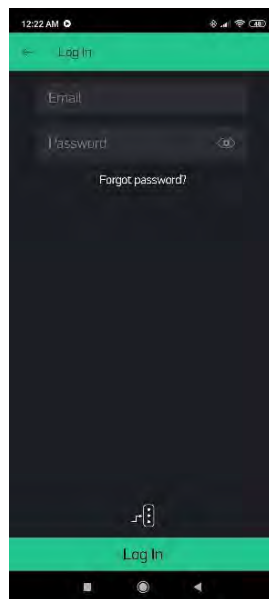


Figura 3.20 Inicio de sesión de la aplicación Blynk.

Una vez dentro de la aplicación se mostrará una pantalla como la de la Figura 3.21, en esta pantalla se selecciona la opción de nuevo proyecto <<New Project>>.

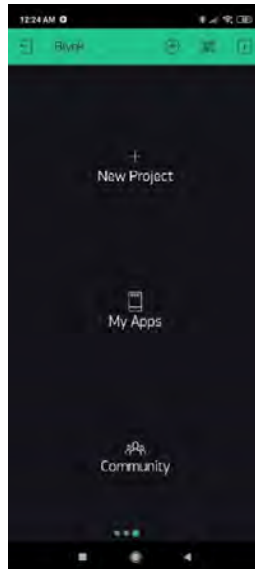


Figura 3.21 Parte uno: Creación de un nuevo proyecto

seguido aparecerá otra pantalla para seleccionar el dispositivo o sistema embebido con el que se va a trabajar en este caso se seleccionó el sistema embebido NodeMCU *ESP8266* después de haber seleccionado el dispositivo con el que se va a trabajar y haber creado un nombre para el proyecto, se selecciona el botón de crear <<Create>>, como se muestra en la Figura 3.22.

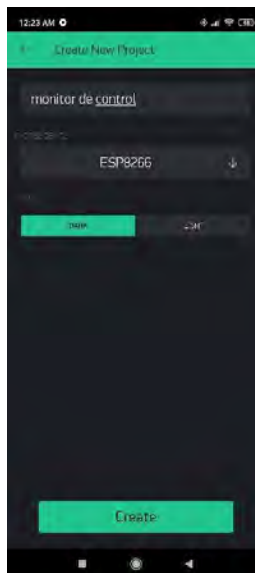


Figura 3.22 Parte dos: Creación de un nuevo proyecto.

Seguido aparecerá un cuadro de texto en la pantalla de la aplicación que muestra un texto diciendo que un token de autenticación fue enviado a nuestro correo electrónico que está vinculado con la cuenta de la aplicación, como se muestra en la Figura 3.23. El token de autenticación que se envió al correo es muy importante ya que, con este el programa que se hizo en la plataforma de Arduino y la aplicación se vinculan para poder enviar y recibir los datos de los sensores por medio de una red WIFI.

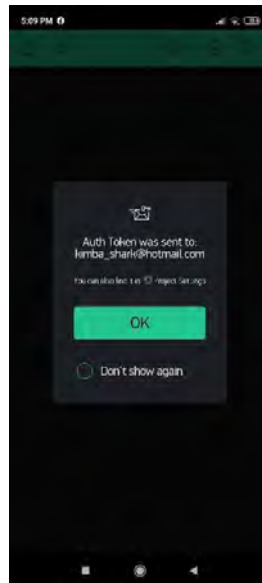


Figura 3.23 Texto de envío de token de autenticación.

Una vez creado un nuevo proyecto se mostrará una pantalla como en la Figura 3.24, en esta selecciona la opción agregar que está en la parte superior derecha de la pantalla con un icono de <<más encerrado en un círculo>>, como se muestra en la Figura 3.24 con el icono encerrado en un cuadro de color rojo.

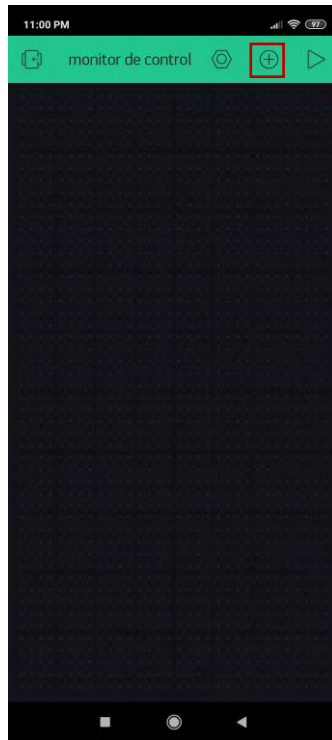


Figura 3.24 Agregar elementos a la interfaz gráfica.

Después de seleccionar el icono de más que aparecerá una ventana como se muestra en la Figura 3.25, en esta ventana nos dirigimos hasta la sección de Displays, en esta sección se seleccionó la opción de **<<Labeled Value>>** con esta opción se agregaron a la interfaz dos pantallas virtual que muestran el valor de temperatura del sensor LM35 y el valor de temperatura del sensor de radiación, de igual manera se seleccionó la opción de LCD, esta agrega una pantalla LCD que mostrara el valor de la señal de control PWM y la temperatura a la que se quiere llegar que son 25 °C, en la Figura 3.25 se puede observar los elementos descritos anteriormente y agregados en la interfaz gráfica de la aplicación.

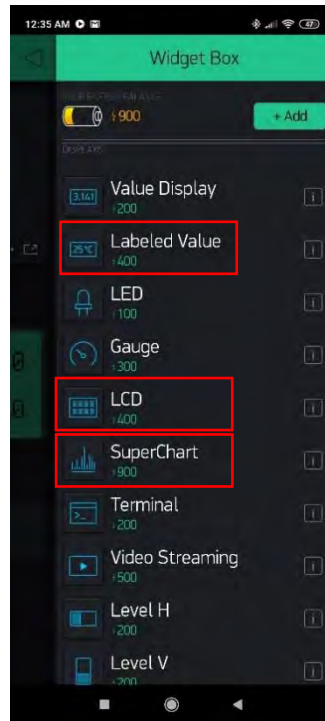


Figura 3.25 Elementos agregados a la interfaz gráfica.

En la sección de “**DISPLAYS**” se seleccionó la opción <<**SuperChart**>> el cual agrega una pantalla donde se pueden observar las gráficas de la temperatura que está midiendo el sensor LM35, la temperatura a la que se quiere llegar y la señal de control PWM, con esta característica agregada se puede exportar un archivo en formato CSV con los valores de las gráficas plasmadas en la aplicación solo que esta tiene una limitante que las muestras que se guardan en el archivo lo hace cada minuto. En la Figura 3.26 se muestra una captura de pantalla donde se observan las opciones de exporta los valores en un archivo csv y otra opción para borrar los datos de las gráficas.



Figura 3.26 Exportación de datos sensados en archivo CSV.

Si se selecciona la opción de exporta a un archivo CSV, está la envía al correo electrónico vinculado a la cuenta de la aplicación, como se puede apreciar en la Figura 3.27.

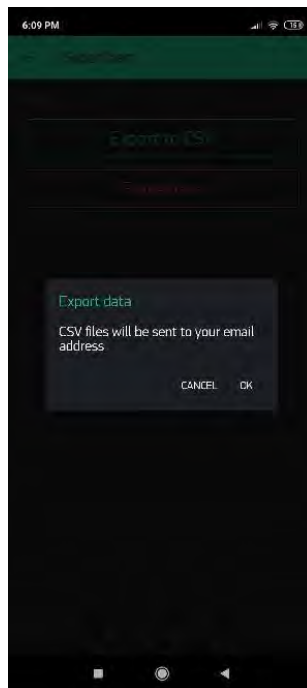


Figura 3.27 Envío del archivo de datos al correo electrónico vinculado con la cuenta.

En la Figura 3.28 se puede apreciar el resultado final de la interfaz gráfica construida en la aplicación Blynk.

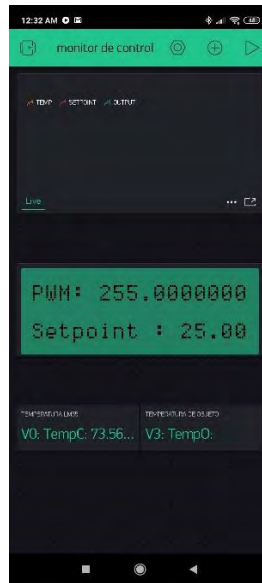


Figura 3.28 Interfaz gráfica creada en la aplicación Blynk.

Código de programa.

En la primera parte del código se puede observar las librerías que se están utilizando para el funcionamiento con el sistema embebido *ESP8266*, como se muestra en la Figura 3.29.



Figura 3.29 Librerías utilizadas en el sistema embebido NodeMCU ESP8266.

seguido en la Figura 3.30 se muestra la configuración de la red a la que se conectara el sistema embebido NodeMCU *ESP8266* al igual se puede observar el token de autenticación que la aplicación envió al correo vinculado con la cuenta de la plataforma Blynk.

```
char auth[] = "phsaqhK8m_T807rnG4PEq8k4IurcS4TZ";  
  
char ssid[] = "Linksys32";  
char pass[] = "A82E512BC";
```

Figura 3.30 Configuración de conexión a una red WIFI y token de autenticación.

En la Figura 3.31 se muestra la parte del código donde se inicia el monitor serial y la conexión con la red y con la aplicación Blynk por medio del Token de autenticación configurada anteriormente como se mostró en la Figura 3.30.

```
void setup()  
{  
  // Debug console  
  Serial.begin(9600);  
  
  Blynk.begin(auth, ssid, pass);  
}
```

Figura 3.31 Inicio de conexión de red y monitor serial.

En la última sección del código **<<void Loop()>>** como se muestra en la Figura 3.32, la configuración de puertos virtuales, la función de estos permite enviar la información a la aplicación del teléfono celular, de los sensores de temperatura, de radiación térmica o cualquier otra variable que se quiere plasmar en los elementos agregados en la interfaz gráfica de la aplicación.

```

void loop()
{
    Blynk.virtualWrite(V0, Input);
    Blynk.virtualWrite(V1, Setpoint);
    Blynk.virtualWrite(V2, Output);
    Blynk.virtualWrite(V3, mlx.readObjectTempC());

    Blynk.run();
}

```

Figura 3.32 Configuración de puertos virtuales para enviar datos a la aplicación Blynk.

3.8 Módulo de lectura y escritura de tarjetas microSD [14].

El módulo para tarjetas de memoria microSD permite manipular el uso de una tarjeta de memoria microSD con las plataformas Arduino, Raspberry Pi y NodeMCU *ESP8266* por medio de programación, con este módulo se puede leer y escribir información en la tarjeta microSD por medio del estándar Bus de interfaz de periféricos serie o bus SPI del inglés (Serial Peripheral Interface), este estándar es usado principalmente para la transferencia de información entre circuitos integrados en equipos electrónicos. En la Figura 3.33 se muestra el módulo de tarjetas microSD con sus pines de conexión.

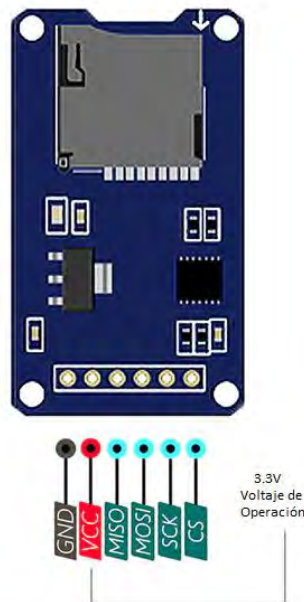


Figura 3.33 Módulo lector de microSD [15].

3.8.1 Modo de operación del módulo microSD.

El SPI es un protocolo síncrono. La sincronización y la transmisión de datos se realiza por medio de las siguientes cuatro señales:

- **SCLK** (*Clock*): Es el pulso que marca la sincronización de datos de transmisión por el NodeMCU *ESP8266*.
- **MOSI** por sus siglas en inglés (*Master Output Slave Input*): Salida de datos del dispositivo Maestro y entrada de datos al dispositivo Esclavo.
- **MISO** por sus siglas en inglés (*Master Input Slave Output*): Salida de datos del dispositivo Esclavo y entrada al dispositivo Master.
- **CS**: selecciona si el dispositivo está en modo Esclavo, o si está en modo Maestro y le indique al dispositivo esclavo cuando se active.

La Cadena de bits se envía de manera síncrona con los pulsos del reloj, esto quiere decir que, con cada pulso del reloj el dispositivo seleccionado como Maestro envía un bit. Para que la transmisión empiece el dispositivo maestro envía una señal al SSTE o SS/Select, con esto el esclavo se activa y empieza la transmisión de datos.

3.8.2 Características del módulo microSD.

- El módulo (adaptador de tarjeta MicroSD) es un módulo lector de tarjetas Micro SD para lectura y escritura a través del sistema de archivos y el controlador de interfaz SPI.
- Admite tarjeta Micro SD, tarjeta Micro SDHC (tarjeta de alta velocidad)
- alimentación de 4.5V ~ 5.5V, 3.3V voltaje regulado por la placa del circuito.
- La interfaz de comunicaciones es una interfaz SPI estándar
- Puertos de alimentación y comunicación (GND, VCC, MISO, MOSI, SCK, CS), GND a tierra, VCC es la fuente de alimentación, MISO, MOSI, SCK para el bus SPI, CS es el pin de señal de selección de chip;
- Conector de tarjeta microSD, fácil inserción de la tarjeta.

3.8.3 Código de programa.

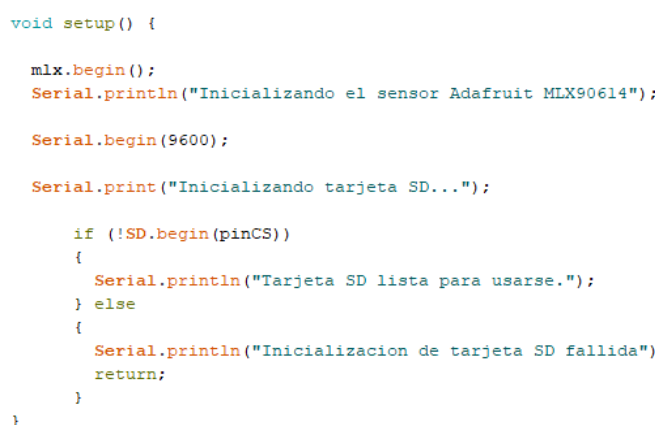
en la Figura 3.34 se puede observar la primera parte de Código donde se agregan las librerías que se utilizaran para el funcionamiento del módulo para tarjetas microSD y el sensor de radiación MLX90614, igualmente se puede observar las variables y objetos utilizados a lo largo del código.



```
Datalogger$  
  
#include <SPI.h>  
#include <SD.h>  
  
#include <Wire.h>  
#include <Adafruit_MLX90614.h>  
  
#define PIN_INPUT A0  
  
Adafruit_MLX90614 mlx = Adafruit_MLX90614();  
const int pinCS = 15;  
  
int Input;  
  
File myFile;
```

Figura 3.34 Librerías y variables de inicio para código de módulo de tarjetas microSD.

En esta parte del código como se puede observar en la Figura 3.35, se muestra la inicialización del sensor de radiación MLX90614 y del monitor serial, al igual se puede apreciar la comprobación de funcionamiento e inicialización del módulo para tarjetas microSD.



```
void setup() {  
  
    mlx.begin();  
    Serial.println("Inicializando el sensor Adafruit MLX90614");  
  
    Serial.begin(9600);  
  
    Serial.print("Inicializando tarjeta SD...");  
  
    if (!SD.begin(pinCS))  
    {  
        Serial.println("Tarjeta SD lista para usarse.");  
    } else  
    {  
        Serial.println("Inicializacion de tarjeta SD fallida");  
        return;  
    }  
  
}
```

Figura 3.35 Inicialización del sensor MLX90614 y modulo microSD.

En la última parte del código en el ciclo void Loop como se muestra en la Figura 3.36, se puede observar la parte del código que corresponde a la lectura del sensor de temperatura LM35 y al sensor de radiación MLX90614 seguido se imprimen los datos de los sensores en el monitor serial de la interfaz del ID de Arduino.

Por último, se crea un archivo con el nombre de "log_data_sd.txt" en este se almacenarán los datos capturados de los sensores o cualquier otra variable que se quiere capturar.

```
void loop(){
  for (int A = 0; A <= 29; A++) {
    Input = Input + analogRead(PIN_INPUT);
  }

  Temp = Input * 100 * 3.3 / 1023;    // Pasar Input a °C
  TempC = Input / 29;

  Serial.print("Temp de objeto °C : ");
  Serial.print(mlx.readObjectTempC());

  Serial.print(" TempC : ");
  Serial.print(TempC);

  myFile = SD.open("/log_data_sd.txt", FILE_WRITE);

  if (myFile) {

    myFile.print("Temp de objeto °C : ");
    myFile.print(mlx.readObjectTempC());

    myFile.print(" TempC : ");
    myFile.print(TempC);

  }
  else {
    Serial.println("error opening datalog.txt");
  }
}
```

Figura 3.36 Lectura y escritura y creación de archivo para almacenamiento de datos en la microSD.

3.9 Diagrama de flujo del programa.

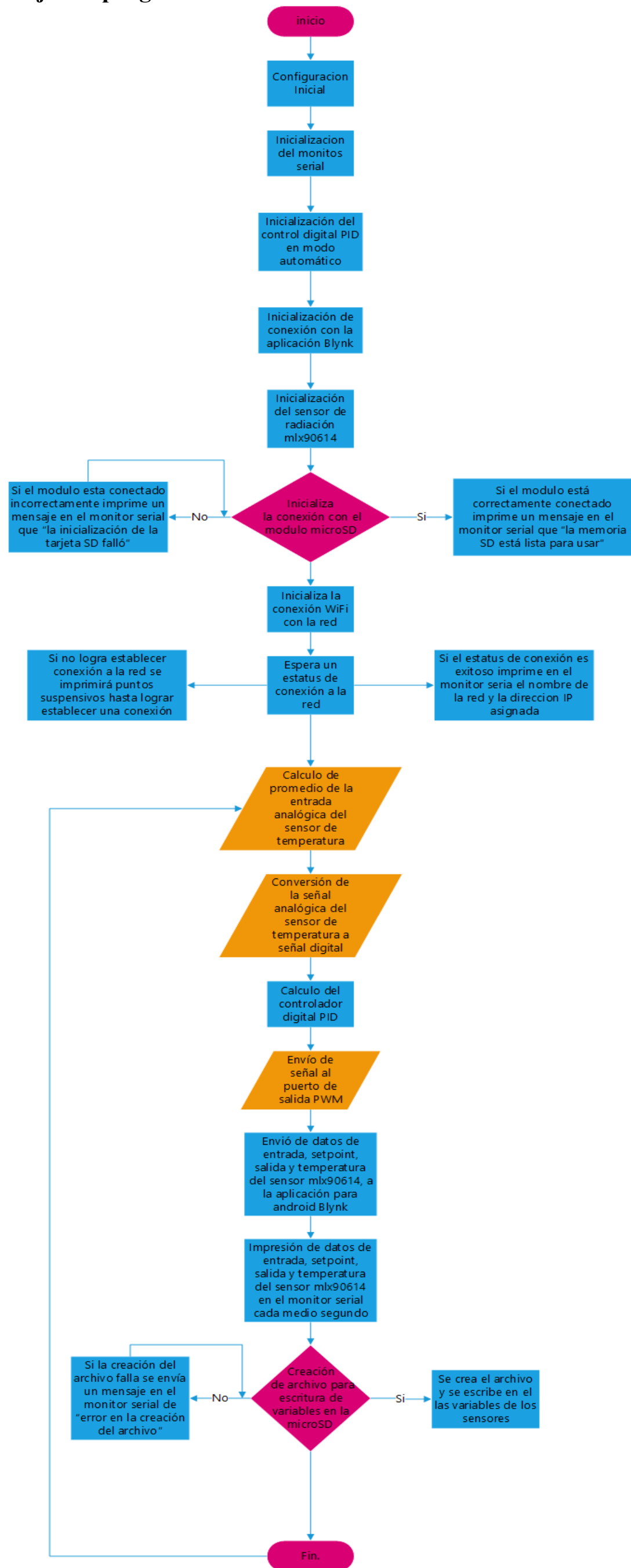


Figura 3.37 Algoritmo del programa para el sistema de control de temperatura.

Capítulo 4 Pruebas experimentales.

4.1 Banco de pruebas.

En este capítulo se mostrará el desarrollo del circuito del sistema de control y las múltiples pruebas que se hicieron a través del proyecto para determinar el correcto funcionamiento del sistema de control de temperatura para el control de la celda Peltier.

4.1.1 Prueba uno: control de temperatura con puente “H” con el sistema embebido Arduino UNO.

Se hicieron múltiples pruebas con el sistema embebido Arduino UNO para determinar correcto funcionamiento de la celda Peltier y poder controlar la temperatura deseada, una de las pruebas fue realizada con un puente H L298 y un potenciómetro para poder regular la temperatura de la celda por medio de una señal PWM una de las entradas analógicas del microcontrolador.

En un principio el sistema parecía funcionar perfectamente, pero al paso de algunos segundos, después de configurar la fuente de poder a 12V y regular la corriente a 1.2A por medio del potenciómetro la corriente se truncaba a 0.8A, al analizar porque estaba ocurriendo el truncamiento de corriente, se analizó que la corriente de la celda estaba consumiendo más de lo que podía soportar el puente H L298 de tal forma que se tuvo que descartar el circuito con el puente H ya que con este problema detectado el sistema construido no era eficiente. En la Figura 4.1 se puede apreciar el primer circuito eléctrico del sistema de control de temperatura controlado manualmente por un potenciómetro.

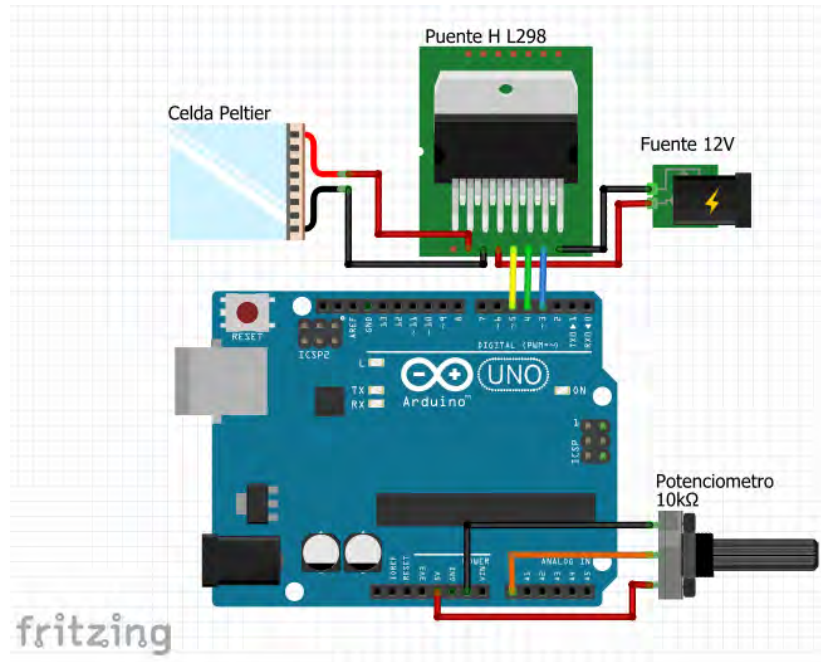


Figura 4.1 Circuito de control de temperatura manual con puente H L298.

4.1.2 prueba dos: control de temperatura con transistor BJT TIP31C a lazo abierto con el sistema embebido Arduino UNO.

Otra de las pruebas que se hizo en el sistema embebido Arduino UNO, fue reemplazando el puente H L298 por un transistor BJT TIP31C, este transistor BJT como se mencionó en el capítulo anterior puede soportar un voltaje máximo de hasta 100V y una corriente máxima de 3A. En esta prueba se pudo observar un buen funcionamiento del sistema ya que no hubo caídas en el voltaje y corriente en la fuente de alimentación, el sistema pudo soportar 15V de entrada en el colector del transistor y variar la corriente de la celda Peltier hasta 2.2A sin problemas. En la Figura 4.2 se puede observar el sistema de control de temperatura a lazo abierto.

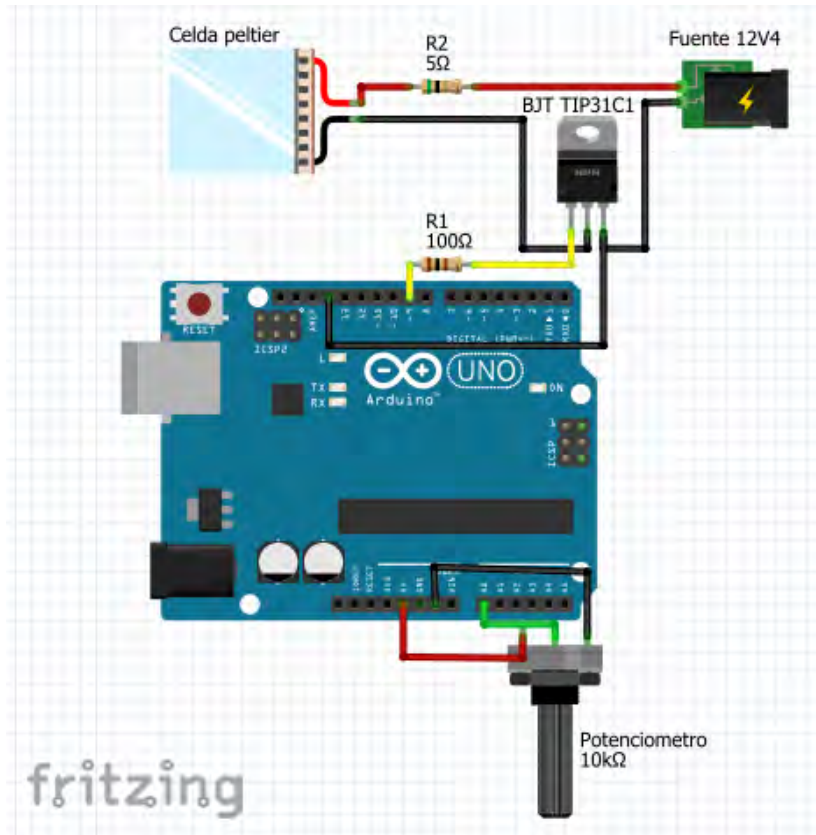


Figura 4.2 Circuito controlador de temperatura con BJT TIP31C a lazo abierto.

4.1.3 prueba tres: control de temperatura con transistor BJT TIP31C a lazo cerrado con el sistema embebido Arduino UNO.

Para las pruebas de funcionamiento del circuito electrónico del sistema de control de temperatura a lazo cerrado se hizo el diseño de un programa en el sistema embebido Arduino UNO, el programa es similar al programa que se hizo en el sistema embebido NodeMCU *ESP8266* y descrito en el capítulo anterior de este proyecto con la diferencia que el microcontrolador Arduino UNO no tiene conexión a internet por lo que se omite la programación que permite la conexión a una red WIFI para el monitoreo y visualización de datos a través de la aplicación para dispositivos celulares llamada Blynk.

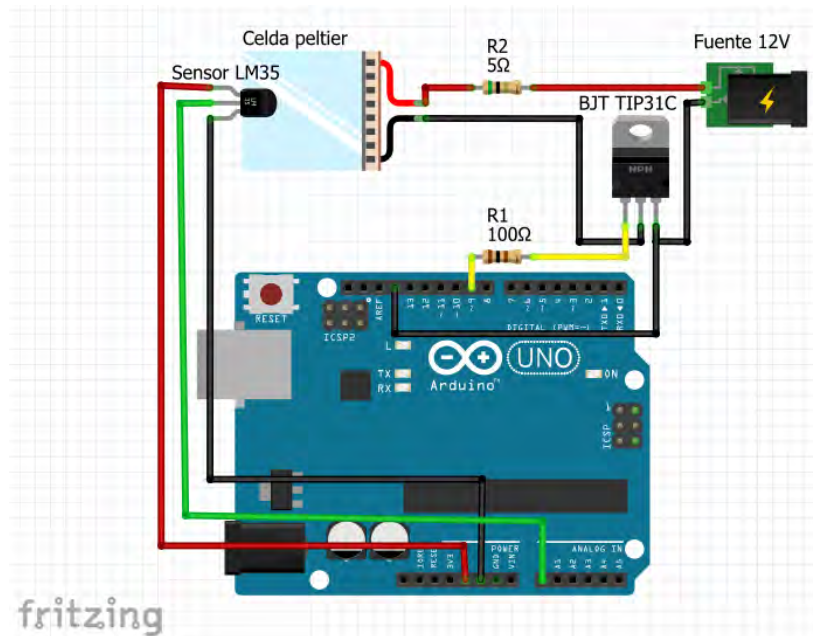


Figura 4.3 Circuito controlador de temperatura con BJT TIP31C a lazo Cerrado.

4.1.4 Prueba cuatro: código del programa en el sistema embebido Arduino Uno.

Una vez comprobado que el sistema de cambio de temperatura manual funcionaba correctamente al variar la señal PWM a través del potenciómetro, se prosiguió reemplazando el potenciómetro por un sensor de temperatura LM35 para crear un sistema de control a lazo cerrado

Como se mencionó anteriormente el circuito a lazo cerrado consiste en un control de temperatura PID digital capaz de modular la señal PWM automáticamente y así mantener una temperatura de 25 °C estables para el correcto funcionamiento del sensor de radiación (termopila), en la Figura 4.3 se muestra el circuito a lazo cerrado donde el sensor de temperatura tiene que estar pegada a la celda Peltier para que tenga una medición precisa y el control pueda funcionar correctamente.

En la Figura 4.4 se puede observar en el recuadro de color rojo las declaraciones de librerías que se usaron en este código, en el recuadro azul de puede observar las variables y puertos utilizados, en el recuadro de color anaranjado se muestra el código de creación de objetos que contiene el código prediseñado de las librerías y

por último en el recuadro de color verde se observa el código de inicialización de los objetos creados con anterioridad.

```
sketch_may22a$  
#include <PID_v1.h>  
#include <SD.h>  
  
#define PIN_INPUT A0 // Sensor de temperatura  
#define PIN_OUTPUT 16 // Salida PWM  
  
double Setpoint, Input, Output; // Variables del PID  
  
double x, y, z; // Variables para el Ardu Graficador  
  
double Kp = 30, Ki = 55, Kd = 0; // Constantes del PID  
  
PID myPID(&Input, &Output, &Setpoint, Kp, Ki, Kd, REVERSE); // Seteos del PID_v1.h  
  
File myFile;  
int pinCS = 15; // Pin  
  
void setup()  
{  
  myPID.SetMode(AUTOMATIC); // Inicio del PID_v1.h  
  Setpoint = 20; // Fijado en 25°C  
  Serial.begin(9600);  
}
```

Figura 4.4 Librerías y variables de inicio del programa.

En la Figura 4.5 se puede observar la parte faltante del código para el sistema embebido arduino UNO, en el recuadro de color rojo se puede ver el código de lectura de temperatura del sensor LM35, en el recuadro azul se muestra el código del cálculo de salida del control digital PID y lo envía a un puerto de salida en forma de señal PWM, en el siguiente recuadro color verde se muestra el código el cual imprime en el monitor serial las variables de entrada, la temperatura que se quiere controlar y la señal de salida PWM para poder visualizarlas y poder monitorizar los valores leídos por el sistema, en el último recuadro que esta de color anaranjado se muestra el código de configuración del módulo para tarjetas microSD para el almacenamiento de los datos de las variables de entrada, temperatura que se quiere controlar y la señal de salida PWM y así almacenarlos en una tarjeta de memoria microSD.

```

void loop()
{
  for (int A = 0; A <= 29; A++) {
    Input = Input + analogRead(PIN_INPUT);
  }

  Input = Input / 30;          // Input entrada de la temperatura (Promediada)
  Input = Input * 100 * 3.3 / 1023; // Pasar Input a °C

  myPID.Compute();           // Calcular el Output
  analogWrite(PIN_OUTPUT, Output); // Escribir el PWM
  delay(1000);              // Retardo

  Serial.print("Input : ");
  Serial.print(Input);

  Serial.print(" Setpoint : ");
  Serial.print(Setpoint);

  Serial.print(" Output pwm : ");
  Serial.println(Output);

  myFile = SD.open("log_temp.csv", FILE_WRITE);
  if (myFile) {

    myFile.print("Input : ");
    myFile.print(Input);

    myFile.print(" Setpoint : ");
    myFile.print(Setpoint);

    myFile.print(" Output pwm : ");
    myFile.println(Output);

  }

  else {
    Serial.println("error opening test.txt");
  }
}
}

```

Figura 4.5 Cuerpo completo de programa de control de temperatura a lazo cerrado.

4.2 Migración de componentes electrónicos del sistema embebido Arduino Uno al sistema embebido NodeMCU ESP8266.

Una vez terminadas las pruebas con el programa para el sistema embebido Arduino UNO y revisar que el sistema del circuito electrónico a lazo cerrado responde correctamente, los componentes electrónicos se migraron al sistema embebido NodeMCU *ESP8266* ya que una de las ideas principal de este proyecto es poder visualizar y monitorear la información dada por los sensores del sistema de control de temperatura por medio de conexión a internet y esto es posible ya que este sistema embebido cuenta con un módulo WIFI integrado y poder establecer conexión inalámbrica a internet.

En esta última versión del sistema de control de temperatura se integró el sensor termopila MLX90614 el cual como ya se mencionó en el capítulo anterior es el sensor principal por controlar y el encargado de medir la radiación térmica de cualquier superficie u objeto que emita dicha radiación. En la Figura 4.6 se muestra un diagrama esquemático con todas las conexiones y componentes del sistema de control de temperatura.

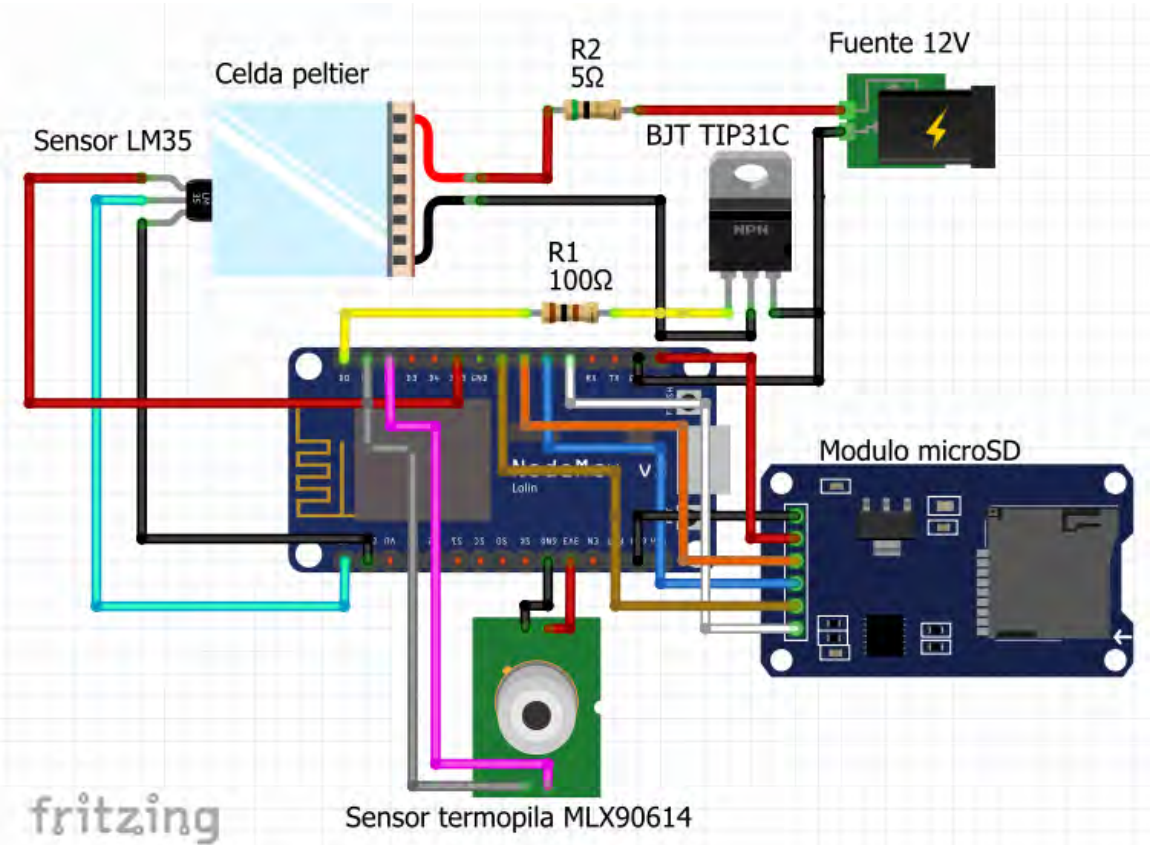


Figura 4.6 Sistema de control migrado al sistema embebido NodeMCU ESP8266.

4.3 Mediciones.

En la Figura 4.7, Figura 4.8 y Figura 4.9 se puede apreciar el prototipo construido con el sistema embebido NodeMCU *ESP8266* y el todo el equipo utilizado a lo largo de este proyecto de tesis para realizar las pruebas y sacar resultados lo resultados experimentales.

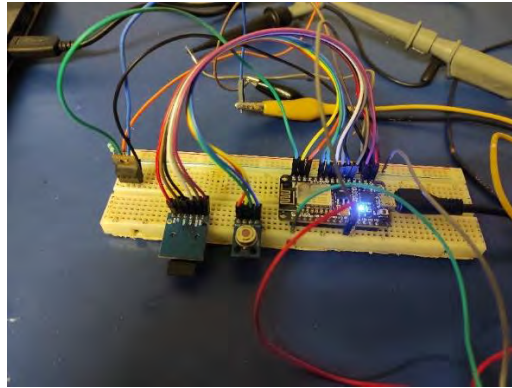


Figura 4.7 Prototipo del sistema de control de temperatura.

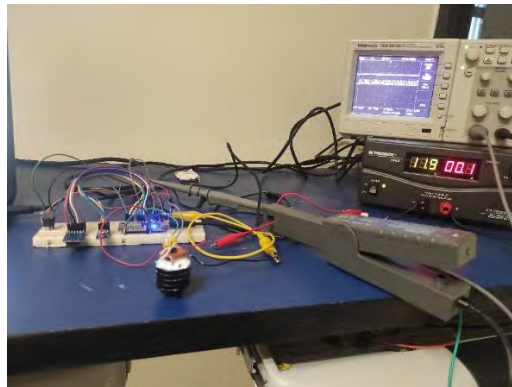


Figura 4.8 Dispositivos y componentes utilizados para el diseño del sistema de control de temperatura.

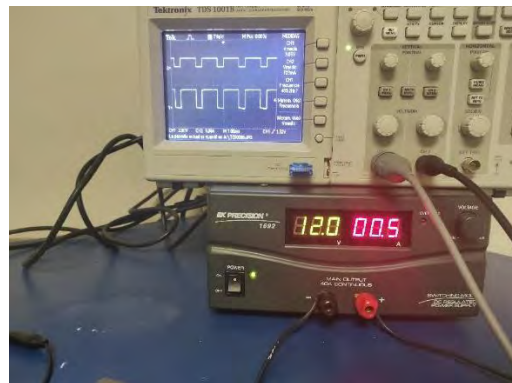


Figura 4.9 Fuente de alimentación y osciloscopio utilizados para el sistema de control de temperatura.

4.3.1 Equipo utilizado

- Osciloscopio Tektronix TDS1001B de dos canales.
- Cámara termográfica FLUKE Ti10.
- Punta de medición de corriente FLUKE 80i-i110s para corriente alterna y corriente directa.
- Sistema embebido NodeMCU ESP-8266.
- Sensor de temperatura termopila MLX90614.
- Sensor de temperatura LM35.
- Celda Peltier 1cm x 1cm.
- Transistor BJT TIP31C.
- Fuente de alimentación variable BK PRECISION 1692.
- Modulo lector de tarjetas microSD.
- Resistencia de 100Ω.
- Resistencia de potencia de 5Ω.
- Protoboard.
- Cables jumper para conexión de Protoboard.

4.3.2 medición de temperatura regulada a 25 °C por medio de control PID.

En la Figura 4.10 se puede apreciar una foto tomada con la cámara térmica FLUKE Ti10 en la que, la temperatura ambiente en la que se encuentra la celda Peltier es de 28 °C.



Figura 4.10 Temperatura ambiente de celda Peltier.

En la Figura 4.11 se puede observar en el canal uno del osciloscopio la medición de una señal de 3.32V, esta es la señal PWM mandada por el microcontrolador a su puerto de salida, con este valor medido se puede confirmar que el sistema de control de temperatura está funcionando correctamente desde un inicio ya que los 3.32V demuestran que el sistema está enviando una señal del 100% de su ciclo de trabajo que es equivalente a los 3.32V que se está midiendo en el canal uno del osciloscopio.

De igual forma en la Figura 4.11 se observa funcionando el canal dos del osciloscopio en este canal se está midiendo la corriente de consumo de la celda Peltier que es de unos 850mA esta es la corriente máxima que puede alcanzar la celda como medida de protección y evitar que se quemara el dispositivo, con esa cantidad de corriente es suficiente para poder alcanzar la temperatura adecuada para que el sistema de control funcione correctamente. Al igual que los 3.32V los 850mA de corriente máxima es equivalente al 100% del ciclo de trabajo de la señal PWM.

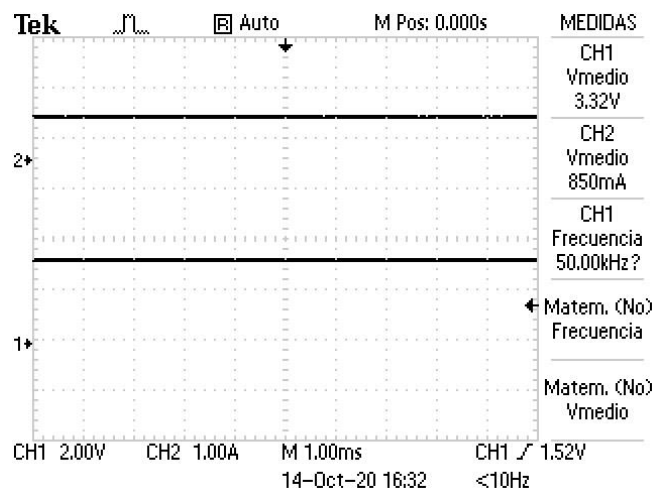


Figura 4.11 Medición de señal PWM y corriente del sistema de control de temperatura a temperatura ambiente.

En la Figura 4.12 se muestran cuatro mediciones obtenidas por medio del monitor serial del ID de Arduino, “Tem de objeto”, “Input”, “Setpoint”, “Output PWM”, en la medición de la variable “Input” se puede ver el valor de 27.53 °C este es la temperatura ambiente en la que se encuentra la celda Peltier al momento de iniciar el control de temperatura, en la Figura 4.10 que se mostró anteriormente se puede

apreciar una temperatura de 28.5 °C la cual tiene 1 °C de error por encima de la temperatura media por el sensor LM35 que está pegado a la celda Peltier. La variable “Setpoint” muestra la temperatura que se quiere alcanzar para que el radiómetro óptico funcione correctamente, por último, en la variable “Output PWM” se observa el valor de 255 este representa la señal PWM en números de bits del microcontrolador en este caso por configuración y para que el sistema funcione correctamente se hizo que trabajara a 8bits que es equivalente a los 255 que muestra en el monitor serial .



Figura 4.12 Medición del sistema de control de temperatura en el ID de Arduino.

4.3.2.1 Medición de temperatura con cámara térmica FLUKE Ti10

En la Figura 4.13 y Figura 4.14 se pueden observar las medidas de temperatura de la celda Peltier al alcanzar los 25 °C programados, la variación de temperatura regulada de la celda Peltier va de 24.9 °C a los 25.3 °C, esto quiere decir que el sistema de control de temperatura funciona correctamente teniendo un error mínimo de 0.1 °C por debajo de los 25 °C y de 0.3 °C por encima de los 25 °C, lo cual, corresponde a la diferencia entre el valor deseado (programado) y la medición de la cámara termográfica.

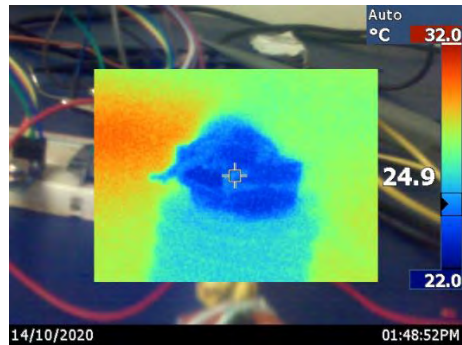


Figura 4.13 Temperatura regulada a 25 °C con error de 0.1 °C por debajo de lo deseado.

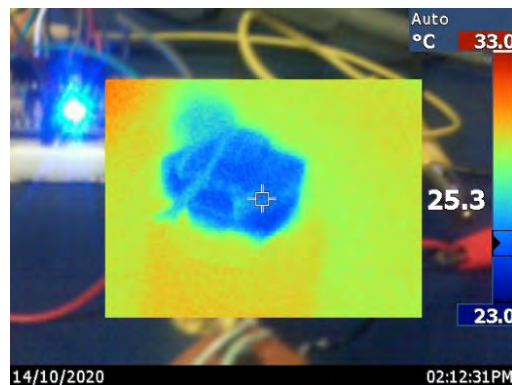


Figura 4.14 Temperatura regulada a 25 °C con error de 0.3 °C por encima de lo deseado.

4.3.2.2 Comportamiento de la señal PWM y corriente consumida por la celda Peltier

En la Figura 4.15 se puede observar la pantalla del osciloscopio la cual está conectado para medir la señal de salida PWM en el canal uno, la señal cuadrada que muestra el canal uno de 903mV es equivalente a los 25.3 °C que muestra la Figura 4.14. El canal dos del osciloscopio de la Figura 4.15 se muestra la corriente de consumo de la celda Peltier, en la señal obtenida de corriente del canal dos del osciloscopio se puede observar una medición de 330mA que al igual a la señal medida del canal uno esta representa los 25.3 °C que se pudo observar en la Figura 4.14.

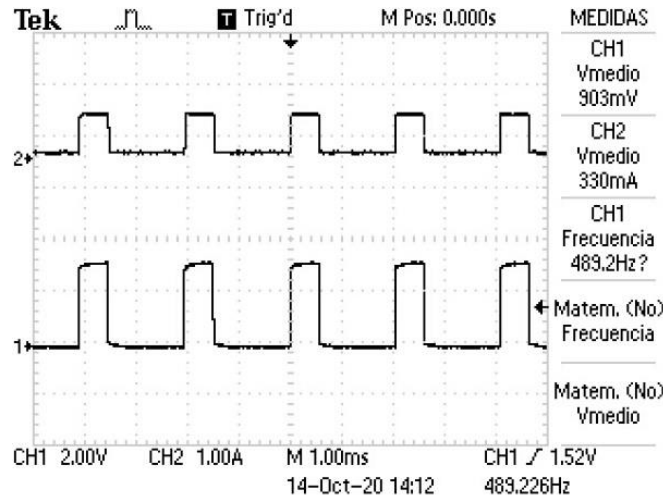


Figura 4.15 Medición de señal PWM y corriente del sistema de control de temperatura a regulada a 25.3 °C.

En la Figura 4.16 se observar dos señales similares a las antes descritas, en el canal uno se muestra un voltaje de 770mV que es equivalente a los 24.9 °C que se pudo observar anteriormente en la Figura 4.13, en el canal dos del osciloscopio se muestra la medición de corriente de consumo de la celda Peltier de 298mA para mantener la temperatura a 24.9 °C.

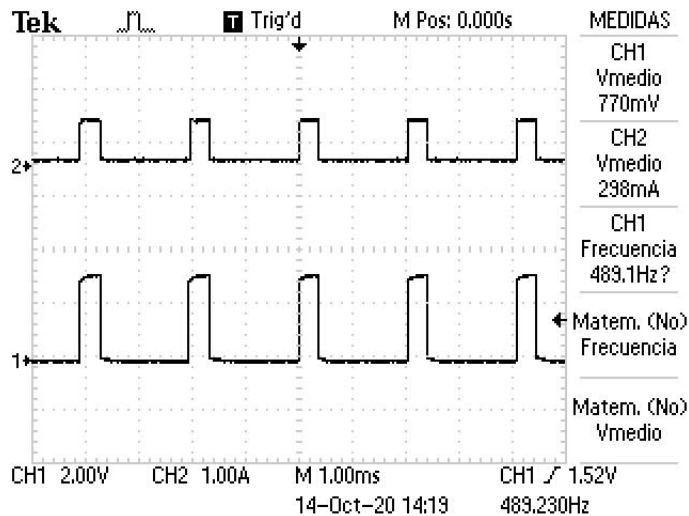


Figura 4.16 Medición de señal PWM y corriente del sistema de control de temperatura a regulada a 24.9 °C.

En la Figura 4.17 se muestran los datos obtenidos en el ID de Arduino del sistema de control de temperatura al alcanzar los 25 °C configurados, se puede observar que en la variable “Input” una temperatura que oscila entre 24.93 °C a 25.28 °C y una señal PWM de 79.24% y 58.35%, estos datos obtenidos vendrían siendo un equivalen a los en voltaje obtenidos en el canal uno del osciloscopio de 903 y 770mV que se muestra en la Figura 4.14 y Figura 4.15, los mismo para las mediciones de corriente que se obtuvieron en el canal dos del osciloscopio de 330mA y 228mA que se muestran en las figuras mencionadas anteriormente en este párrafo.

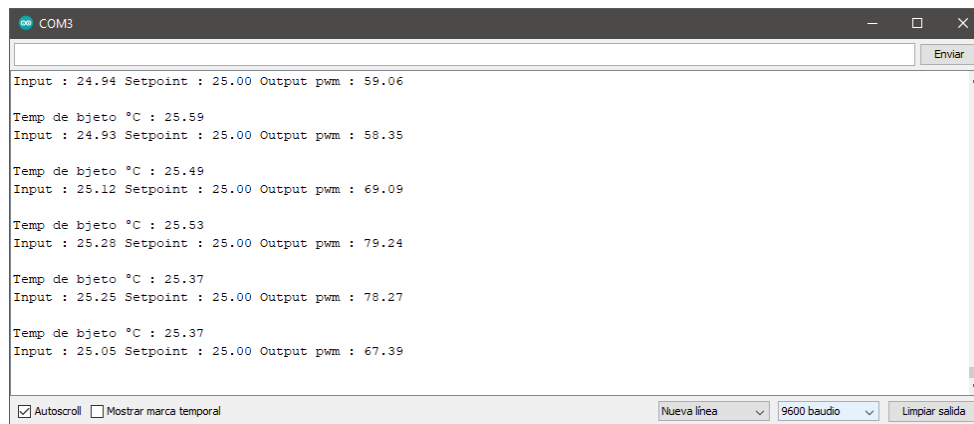


Figura 4.17 Medición del sistema de control de temperatura al alcanzar los 25 °C.

4.4 Conclusiones

El sistema de control y regulación de temperatura para un radiómetro óptico lineal para medición de altos flujos radiativos con apuntamiento preciso cumple satisfactoriamente lo planteado en el proyecto.

Para el control de la temperatura de la celda de Peltier se empleó un circuito Chopper con transistor bipolar y resistencia shunt. Se realizaron varias pruebas en el diseño hasta lograr el control de la celda. Se empleó un sensor de temperatura analógico LM35, sin embargo, esto puede ser mejorado empleando un sensor de temperatura digital como el DS18B20.

El diseño del programa se construyó en el software ID de Arduino el cual puede capturar, visualizar y almacenar datos procesados por la tarjeta *ESP8266*. Se ha configurado el sistema de adquisición de datos para que envíe datos por internet a través de acceso WiFi a los servidores de la aplicación Blynk para el dispositivo móvil con sistema operativo Android, por lo que se pueden enviar y recibir datos de la tarjeta hacia otros dispositivos o viceversa, al igual se instaló un servidor FTP para tener acceso a los archivos de las mediciones hechas por el sistema embebido NodeMCU *ESP8266*. Aún faltan pruebas más exhaustivas para optimizar el diseño, pero los accesos a laboratorios de investigación de la UQROO permanecen cerrados por la contingencia del COVID 19.

Como experiencia personal quiero mencionar la importancia que tuvieron las asignaturas cursadas durante los dos años del posgrado que cursé ya que, gracias a toda la experiencia y conocimiento de los profesores que me dieron clases puede reforzar mis conocimientos y aprender cosas nuevas, para concluir satisfactoriamente este proyecto de tesis.

4.5 Trabajos futuros.

Como trabajo futuro de este proyecto de tesis se puede implementar una aplicación por medio de alguna plataforma para desarrollo de aplicaciones para dispositivos móviles con sistema operativo Android o IOS que permita conectividad a internet

por medio de WIFI para visualización de datos adquiridos de la tarjeta, al igual se puede seguir haciendo pruebas con el servidor FTP para tener un respaldo seguro de los datos adquiridos en caso de que llegara a fallar el almacenamiento en la tarjeta microSD.

Otra aportación que se puede anexar a este trabajo es el diseño e implementación de un circuito electrónico donde se integren los elementos (termopila, sensor de temperatura, celda peltier, transistor y microcontrolador) en una placa PCB con montaje superficial para una mejor organización de los componentes.

Otro trabajo futuro que se puede implementar para que el sistema de control sea más eficiente, es utilizando un segundo transistor BJT TIP32 PNP en una arreglo Push-Pull para así poder suministrar potencia a la celda peltier en polaridad positiva o negativa, para enfriar o calentar en forma correspondiente a la celda peltier, para este tipo de arreglo es necesario implementar una fuente bipolar en la etapa de potencia para poder alimentar al transistor TIP32 con voltaje negativo.

Referencias.

- [1] R. Castrejon, “Radiómetro Unidireccional de Amplio Espectro de radiación y de Lectura Directa,” 9600111, 1995.
- [2] A. Pathak And V. Goel, “Heat Pump Design Using Peltier Element For Temperature Control Of The Flow Cell,” *Int. J. Comput. Sci. Eng. Appl.*, Vol. 3, No. 3, Pp. 41–47, 2013, Doi: 10.5121/Ijcsea.2013.3305.
- [3] Z. Slanina, M. Uhlik, And V. Sladeczek, “Cooling Device With Peltier Element For Medical Applications,” *Ifac-Papersonline*, Vol. 51, No. 6, Pp. 54–59, 2018, Doi: 10.1016/J.Ifacol.2018.07.129.
- [4] Melexis, “MLX90614 - Single And Dual Zone Infrared Thermometer In TO-39,” *Microeletronic Integrated Systems*. P. 35, 2006, Doi: 10.4193/Rhino13.145.
- [5] D. M. Rowe, D. Ph, And S. D, *Handbook*, 1st Editio. Boca Raton: Taylor & Francis Group, 2006.
- [6] G. Patterson And M. Sobral, “Efecto Peltier,” No. 2, 2007.
- [7] Wilfredo Blancarte Lizarraga, “Instrumentacion Para El Control De Procesos Industriales,” 2001.
- [8] A. S. Morris And R. Langari, *Measurement And Instrumentation*, Second Edi. 2016.
- [9] A. P. . Sandoval G, E. Espinosa J, And J. L. Barahona A, “Celdas Peltier : Una Alternativa Para Sistemas De Enfriamiento Con Base En Semiconductor .,” 2008, Pp. 1–9.
- [10] A. Al Dahoud And M. Fezari, “Nodemcu V3 For Fast Iot Application Development,” No. October, P. 5, 2018.
- [11] National Instruments Corp., “Sistema De Adquisición De Datos Y Acondicionamiento De Señales.” [Http://Sine.Ni.Com/Np/App/Main/P/Ap/Daq/Lang/Es/Pg/1/Sn/N17:Daq/Fmid/652/](http://Sine.Ni.Com/Np/App/Main/P/Ap/Daq/Lang/Es/Pg/1/Sn/N17:Daq/Fmid/652/).
- [12] D. O. Correa, “CONTROL PID DIGITAL Algoritmo, diseño e implementación,” 2011.
- [13] Technology inc., “TE-17-1.0-2.5 Thermoelectric Module (Peltier Module) Specifications.” pp. 1–7, 2013.
- [14] DATASHEET SD CARD, “Micro SD Card Module for Arduino.” 2013, [Online]. Available: <http://datalogger.pbworks.com/w/file/fetch/89507207/Datalogger - SD Memory Reader Datasheet.pdf>.

- [15] ElectroPeak, “sd card module with arduino,” 2019. <https://create.arduino.cc/projecthub/electropeak/sd-card-module-with-arduino-how-to-read-write-data-37f390>.
- [16] Blynk, “Home.” <https://blynk.io/>.
- [17] H. Durani, M. Sheth, M. Vaghasia, and S. Kotech, “Smart Automated Home Application using IoT with Blynk App,” Proc. Int. Conf. Inven. Commun. Comput. Technol. ICICCT 2018, no. Icticct, pp. 393–397, 2018, doi: 10.1109/ICICCT.2018.8473224.
- [18] F. Peeters, M. Peetermans, and L. Indesteege, “Temperature Sensors,” Mod. Sensors Handb., pp. 347–393, 2010, doi: 10.1002/9780470612231.ch8.
- [19] K. Ogata et al., Ingeniería de control moderna. 2010.
- [20] M. Gutiérrez Hinestroza and S. A. Iturralde Kure, Fundamentos Básicos de Instrumentación y Control. 2017.
- [21] J. O. Aguilar, R. C. García, and O. A. Jaramillo, “Diseño de un sistema electrónico para amplificación y enfriamiento del detector de un radiómetro óptico.”
- [22] J. O. Aguilar and O. A. Jaramillo, “Figuras Radiómetro,” Terra, pp. 1–5, 2007.
- [23] C. Pardo, “Controlador PID,” 2020. <https://www.picuino.com/es/arduprog/control-pid.html>.
- [24] J. I. Hernández Vega, “El Software Embebido y los Retos que Implica su Desarrollo,” 2010, [Online]. Available: <https://www.redalyc.org/articulo.oa?id=94415759009>.

Anexos.

Código completo.

```
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <PID_v1.h>

#include <SPI.h>
#include <SD.h>
#include <Wire.h>
#include <Adafruit_MLX90614.h>

#define PIN_INPUT A0 // Sensor de temperatura

#define PIN_OUTPUT 16 // Salida PWM

char auth[] = "phsaqhK8m_T807rnG4PEq8k4IurcS4TZ";

char ssid[] = "INFINITUMh75e Mecatronica ";
char pass[] = "CostaR1ca";

double Setpoint, Input, Output; // Variables del PID

double Kp = 30, Ki = 55, Kd = 0; // Constantes del PID

PID myPID(&Input, &Output, &Setpoint, Kp, Ki, Kd, REVERSE); // Seteos del PID_v1.h
File myFile;
```

```

int pinCS = 15; // Pin

Adafruit_MLX90614 mlx = Adafruit_MLX90614();

void setup(){

myPID.SetMode(AUTOMATIC); // Inicio del PID_v1.h
Setpoint = 25;           // Fijado en 25°C

Serial.begin(9600);

Blynk.begin(auth, ssid, pass);
mlx.begin();

    // SD Card Initialization
    if (SD.begin(pinCS))
    {
        Serial.println("SD card is ready to use.");
    } else
    {
        Serial.println("SD card initialization failed");
        return;
    }
}

void loop(){
    for (int A = 0; A <= 29; A++) {

```

```

Input = Input + analogRead(PIN_INPUT);
}

Input = Input / 30;          // Input es entrada de la temperatura (Promediada)

Input = Input * 100 * 3.3 / 1023; // Pasar Input a °C

myPID.Compute();          // Calcular el Output

analogWriteFreq(490);
analogWriteRange(255);

analogWrite(PIN_OUTPUT, Output); // Escribir la señal PWM

delay(1000);              // Retardo

Blynk.virtualWrite(V0, Input);
Blynk.virtualWrite(V1, Setpoint);
Blynk.virtualWrite(V2, Output);
Blynk.virtualWrite(V3, mlx.readObjectTempC());

Serial.print("Temp de objeto °C : ");
Serial.print(mlx.readObjectTempC());

Serial.print(" Input : ");
Serial.print(Input);
Serial.print(" Setpoint : ");
Serial.print(Setpoint);

```

```
Serial.print(" Output pwm : ");
Serial.println(Output);

delay(500);

SD.remove("log_data_sd.text");
myFile = SD.open("log_data_sd.text", FILE_WRITE);

if (myFile) {

myFile.print("Tem de objeto °C : ");
myFile.print(mlx.readObjectTempC());

myFile.print("Input : ");
myFile.print(Input);

myFile.print(" Setpoint : ");
myFile.print(Setpoint);

myFile.print(" Output pwm : ");
myFile.println(Output);

}
else {
Serial.println("error opening test.txt");
}
Blynk.run();
}
```