

Instituto Tecnológico de Tuxtla Gutiérrez

“Instrumentación De Un Sensor De Temperatura Por Onda Reflectiva”

ASESOR

Dr. Jorge Luis Camas Anzueto

REVISORES

ALUMNO

Mauricio Ivan Guillen Escobar

**INGENERIA ELECTRONICA
RESIDENCIA PROFESIONAL**

Contenido

CAPÍTULO 1. GENERALIDADES	1
1.1 Introducción	1
1.2 Información de la institución donde se desarrolló el proyecto	2
1.2.1 Historia del ITTG	2
1.2.2 Misión	3
1.2.3 Visión	3
1.2.4 Localización	4
1.2.5 Área específica relacionada directamente con el proyecto	4
1.3 Antecedentes	5
1.4 Planteamiento del problema	5
1.5 Nombre del proyecto	5
1.6 Objetivos	5
1.6.1 Objetivo general	5
1.6.2 Objetivos específicos	5
1.7 Justificación del proyecto.	5
1.8 Delimitación del tema	6
1.9 Metodología para el desarrollo del proyecto.	6
CAPÍTULO 2. FUNDAMENTO TEÓRICO	8
2.1 Estructura del proyecto	8
2.2 transimpedancia	8
2.3 Fuente Simétrica	9
2.4 Transformador de entrada	10
2.5 Rectificador a diodos	10
2.6 El filtro	12
2.7 Filtro con condensador a la entrada	12
2.8 El regulador	12
2.9 Regulador ajustable LM317	14
2.10 Regulador Ajustable de potencia LM350	15
2.11 Etapa de potencia	15
2.12 Programa de apoyo	17
2.12.1 Matlab	17
2.12.2 Vectores de datos	21
2.12.4 Función definida en forma paramétrica	23
2.12.5 Rellenando áreas de color, fill	26
2.12.6 Decorando las gráficas	27
2.12.6.1 legend	27
2.12.6.2 axis	28

2.12.6.3 Autoescalado	29
2.13 Microcontrolador	30
2.13.1 Arduino UNO	30
2.13.2 Estructura básica de un programa	32
2.13.3 Tipos de Datos	32
2.13.4 Funciones de Tiempo	33
2.14 Sensor DS18B20	33
2.15 Laser	34
2.16 Fotodetector	35
2.17 Fibra multimodo	35
2.17.1 Aplicaciones	35
2.17.2 Tipos	36
2.18 Conectores de Fibra multimodo (ST)	37
CAPÍTULO 3. DESARROLLO E IMPLEMENTACIÓN DEL PROYECTO	38
3.1 Diseño del sistema de adquisición de datos	38
3.1.1 Sensor utilizado, DS18B20	38
3.1.2 Celda de peltier	38
3.1.3 Diagrama a bloques	39
3.2 Construcción de la etapa de potencia	39
3.2.1 Celda de peltier y Resistencia Eléctrica	39
3.3 Interface gráfica de MATLAB	41
CAPÍTULO 4. PRUEBAS Y RESULTADOS	42
4.1 Resistencia Eléctrica	42
4.2 Celda de peltier	42
4.3 Amplificador de transimpedancia	43
4.4 Funcionamiento	44
Conclusiones	46
Referencias	47
Anexos	48
A1. DataSheet, OPF 372A	48
A2. DataSheet, OPF 472	51
A4. DataSheet, C358C	54
A5. Programa grabado en el microcontrolador	57
A6. Imágenes Del Proyecto Terminado	61
Apéndice – Manual de operación.	63
A 1.1 Funcionamiento de la fuente de alimentación del diodo láser.	63

A 1.2 Funcionamiento de la fuente simétrica.....	64
A 1.3 Etapas de potencia	65
A 1.4 Sensor de transimpedancia.....	66
A 1.5 interface gráfica con el usuario	67
A 1.6 Descripción del funcionamiento del proyecto	67

CAPÍTULO 1. GENERALIDADES

1.1 Introducción

Igual que el hombre interactúa con los procesos de la naturaleza, modificándolos y guiándolos para su máximo provecho y beneficio, también se propone crear sistemas y arquitecturas capaces de emularle y ayudarle en la toma de decisiones o en su caso, incluso actuar autónomamente. Estas estructuras inteligentes han de ejecutar necesariamente tres funciones básicas: la detección o medida, el procesado y decisión, y finalmente, la actuación. Las dos primeras funciones requieren sensores y es donde se centra este trabajo. En términos generales un sistema sensor se compone de un elemento que genera una señal en relación con el parámetro a medir (transductor), un canal de comunicaciones que transporta la señal generada hasta una unidad que procesa y presenta la información que se pretende y que en definitiva relaciona el sistema sensor con el usuario o con otros sistemas externos a él.

Entre los sistemas sensores se encuentran los “sensores ópticos” y los “sensores de fibra óptica”, donde la magnitud a medir introduce modificaciones a la luz o genera luz, según sea el caso, que se transporta a través de fibra óptica y se detecta para su conversión al dominio eléctrico a través de una unidad optoelectrónica, que se encarga, además, de procesar la señal recibida y extraer la información requerida [1] como se muestra en la figura 0.1.

Los avances impresionantes de la tecnología fotónica como consecuencia de las telecomunicaciones y las características propias de la tecnología de fibra han posibilitado la introducción exitosa de sensores de fibra óptica (fig.1.1), en sectores o “nichos de aplicación” en donde los sensores tradicionales no funcionan correctamente o simplemente no funcionan. Así se están empleando en ambientes con grandes campos electromagnéticos (Ej. Centrales generación eléctrica), o en el entorno donde la generación de señales eléctricas es peligroso. (Ej. Oleoductos, centrales de producción de biogás, aviones), o en aplicaciones donde se precisan sistemas de pequeño tamaño y compatibles con el objeto u organismo a medir (Ej. Sensores biomédicos) o bien en lugares donde la temperatura ambiente es tan elevada que los sensores tradicionales no funcionan adecuadamente debido a múltiples factores (Ej. Acerías, fundiciones, soldadura) [2, 3].

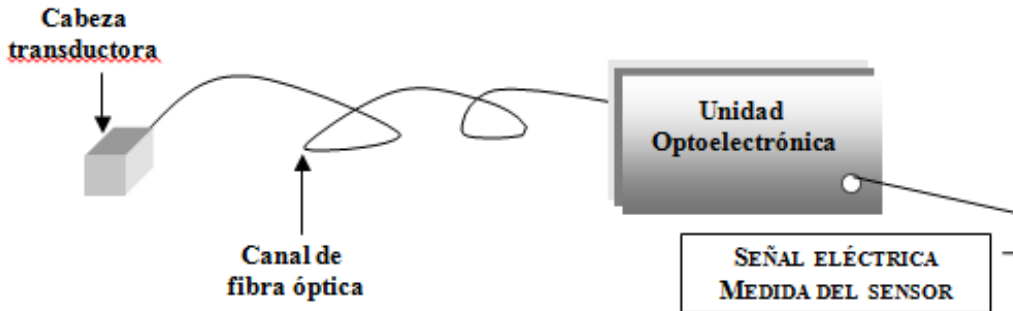


Figura 1.1 Composición genérica de un sensor de fibra óptica

1.2 Información de la institución donde se desarrolló el proyecto

1.2.1 Historia del ITTG

El Instituto Tecnológico de Tuxtla Gutiérrez es una institución pública dependiente de la Secretaría de Educación Pública. Imparte 8 licenciaturas y 2 programas de posgrado en las áreas de Ingeniería, Tecnología y Ciencias Administrativas.

El Instituto Tecnológico de Tuxtla Gutiérrez fue fundado el 24 de octubre de 1972, se encuentra ubicado en el centro del estado de Chiapas, que por su gran variedad de climas y suelos es propio para el cultivo de muy diversas especies vegetales nativas y adaptadas, con lo que se puede desarrollar la agroindustria, así como muchos otros procesos industriales a partir del gran potencial que ofrece este estado, en el cual el Instituto debe constituirse en el sujeto de cambio, al presentar alternativas de desarrollo sustentable del estado en forma multidisciplinaria.

La matrícula escolar en licenciatura se conforma de 1 982 estudiantes: 1 473 hombres y 509 mujeres. En el posgrado hay 78 alumnos: 68 hombres y 10 mujeres.

La población de egresados en el nivel de licenciatura es de 2 345 alumnos, de ellos 1 892 son hombres y 453 son mujeres.

Los principales laboratorios con los que cuenta el Instituto Tecnológico de Tuxtla Gutiérrez son:

- ✓ Microbiología
- ✓ Biotecnología
- ✓ Química
- ✓ Química pesada

- ✓ Mecánica
- ✓ Sistemas computacionales
- ✓ Ingeniería industrial
- ✓ Plantas piloto
- ✓ Polo Tecnológico Nacional

Proyectos de vinculación exitosos con la industria

- ✓ Aprovechamiento integral del timbre
- ✓ Incremento en el contenido de sacarosa en la caña de azúcar
- ✓ Efectos del boro en la producción del cacahuete
- ✓ Factibilidad técnica en la extracción de aceites esenciales de la flora chiapaneca

1.2.2 Misión

Formar de manera integral profesionistas de excelencia en el campo de la ciencia y la tecnología con actitud emprendedora, respeto al medio ambiente y apego a los valores éticos.

1.2.3 Visión

Ser una Institución de excelencia en la educación superior tecnológica del Sureste, comprometida con el desarrollo socioeconómico sustentable de la región.

1.2.4 Localización

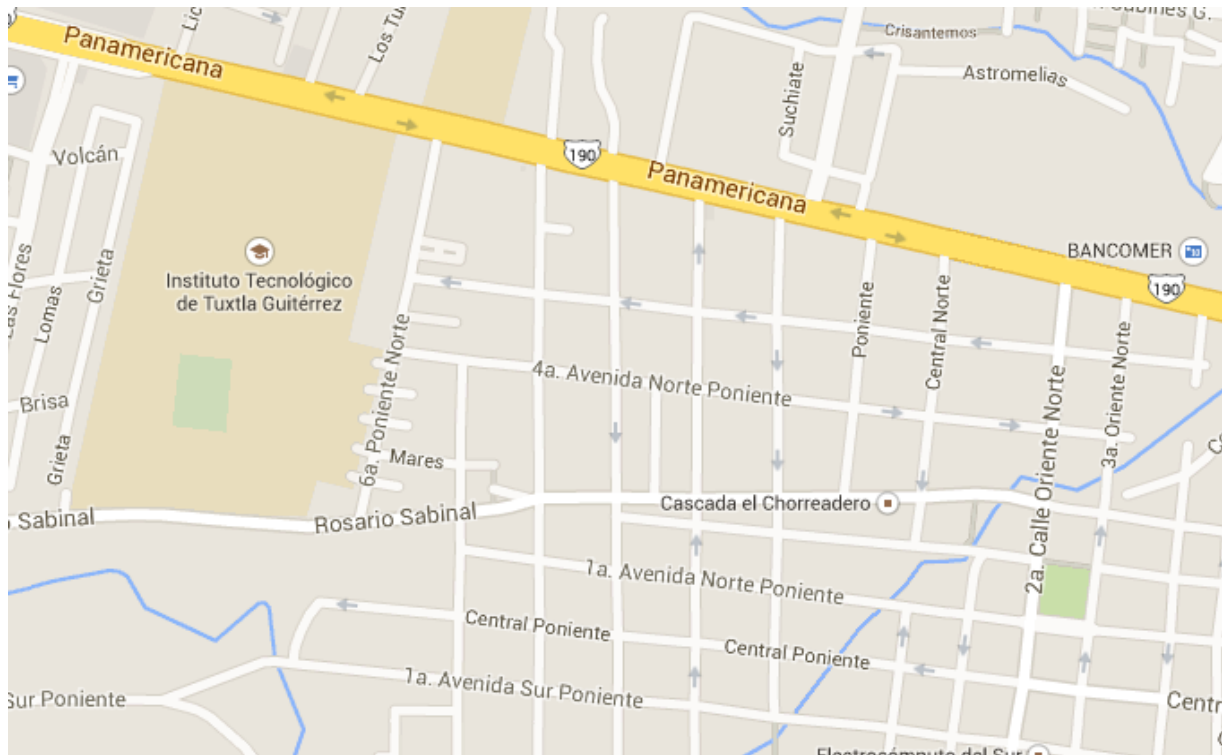


Fig. 1.2 Carretera Panamericana Km 1080. Tuxtla Gutiérrez Chiapas, C.P. 29050.

1.2.5 Área específica relacionada directamente con el proyecto

El laboratorio de Ingeniería Electrónica cuenta con 8 áreas destinadas a la investigación, desarrollo e implementación de proyectos escolares, 5 de las cuales son educativas, en las que los docentes imparten clases de: electrónica digital, PLC, electrónica de potencia, instrumentación, mediciones eléctricas, programación en c y electrónicas analógicas. Dos más son para investigación, una pertenece al grupo de IEEE y la otra es de desarrollo de circuitos electrónicos. La última área está destinada para que los alumnos desarrollen sus actividades en extra clase. Parte del área que se ocupó en la realización de éste proyecto fueron las de IEEE, las de Electrónicas Analógicas y desarrollo de circuitos electrónicos y el almacén de Mecatrónica.

La carrera de Ingeniería Electrónica tiene como misión “formar profesionales de excelencia con competencias en el ámbito de la Ingeniería Electrónica, motivados para la promoción del desarrollo profesional y el conocimiento científico y tecnológico, con actitud emprendedora, respeto al medio ambiente y apego a los valores cívicos y éticos”.

1.3 Antecedentes

En León Guanajuato se realizó una tesis denominada “Caracterización de un sensor de temperatura basado en fibra óptica dopada con Erbio”, en el centro de investigaciones en óptica. Cuyo autor es el Ing. Víctor Eduardo López Padilla, con fecha de realización, septiembre de 2008.

1.4 Planteamiento del problema

Actualmente se cuenta con distintos sensores de temperatura que van desde el rango de sensibilidad de $\pm 1\%$, por lectura, además de los altos precios que hay que pagar, lo que se planea con este proyecto no es más que la mejora y la innovación de un sensor de temperatura basado en fibra óptica.

1.5 Nombre del proyecto

Instrumentación de un sensor de temperatura por onda reflectiva

1.6 Objetivos

1.6.1 Objetivo general

Instrumentar y caracterizar el sensor de temperatura de fibra óptica en un rango de temperatura de 20°C a 90°C.

1.6.2 Objetivos específicos

- Diseñar y construir una fuente de voltaje para el láser emisor de luz.
- Diseñar y construir un cubo de cristal que pueda soportar bajas y altas temperaturas.
- Construir un sistema de acondicionamiento de temperatura.
- Elaborar un programa en arduino para la automatización.

1.7 Justificación del proyecto.

Se pretende mejorar e innovar un sensor de temperatura por fibra óptica, haciéndolo más barato que los establecidos en el mercado, los puntos que hay que mejorar son la fuente de voltaje que alimenta el láser, el empalme en la fibra óptica y el sensor de transimpedancia, por parte de la fuente de luz es necesario que sea lo más estable posible ya que si esta varía afectaría el proceso de caracterización.

En la parte del empalme de la fibra óptica es necesario hacer un empalme limpio con el menor número de pérdidas ya que si existen muchas pérdidas la luz emitida por el láser no llegará completa al receptor.

El impacto sobre el ambiente será menor puesto a que se trabajara con fibra óptica que es un material menos contaminante que los semiconductores que usan los sensores de temperatura actualmente.

También se pretende que los datos arrojados en una hoja de datos Excel sean entendibles para cualquier persona que los vea.

1.8 Delimitación del tema

El proyecto se ha contemplado como un sensor de temperatura basado en fibra óptica, el cual se meterá en un recipiente cubico para mantener monitoreando la temperatura a la que se somete la fibra óptica, a través del censado de la temperatura y los valores obtenidos por el sensor de transimpedancia poder caracterizar el sensor de fibra óptica basado en los resultados obtenidos de pruebas diarias. Cabe mencionar que este proyecto se hará en primer plano como un prototipo.

1.9 Metodología para el desarrollo del proyecto.

❖ DISEÑO Y CARACTERIZACION DE LA FUENTE DE LUZ:

- A) Diseño de la fuente de alimentación del diodo emisor de luz OPF372A con monitoreo de corriente.
- B) Caracterizar la fuente de luz con un fotodetector calibrado.

❖ CARACTERIZACION DEL FOTODETECTOR:

- A) Diseñar el detector de transimpedancia para el fotodetector OPF472
- B) Caracterización del fotodetector con la fuente de luz y relacionar los niveles de potencia con los niveles de voltaje obtenidos del detector de transimpedancia.

❖ CONSTRUCCION DEL SENSOR DE TEMPERATURA:

- A) Para la construcción del sensor de temperatura se conectara la fuente de luz y el detector de transimpedancia a un acoplador de fibra óptica una por 2 canales y se empalmara una fibra óptica que contiene un material termocrónico.

❖ CARACTERIZACION DEL SENSOR:

- A) Se construirá un sistema de calentamiento con un monitoreo de temperatura en un rango de 20°C a 90°C, se insertara el sensor de temperatura en el sistema de calentamiento y se medirá la potencia obtenida por el fotodetector en función de la temperatura.

CAPÍTULO 2. FUNDAMENTO TEÓRICO

2.1 Estructura del proyecto

Cuenta con un deposito cubico que nos sirve para concentra temperatura, el propósito de este cubo es el de mantener encerrada la temperatura y así poder monitorearla con sensores de temperatura ver fig.2.1.

Las medidas del cubo son de 10x10x10 Cm, con un espesor de 4mm y con 2 orificios a los costados con un diámetro de 1mm, además de estar montada a dos barras de aluminio que funcionan como disipadores de calor con una medida de 30cm de largo con una altura de 7cm.

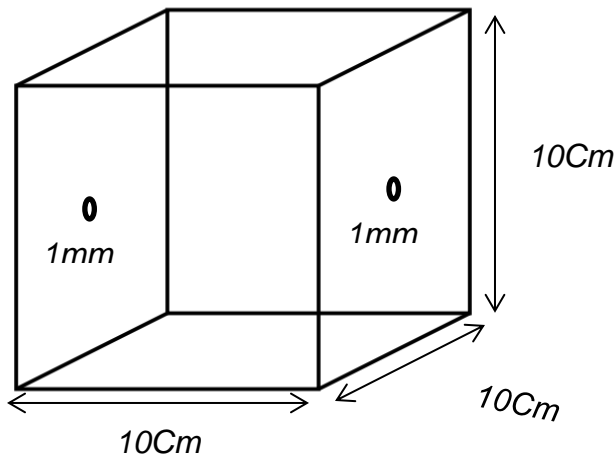


Figura 2.1 Características del cubo para concentrar temperatura

2.2 transimpedancia

Amplificadores de transimpedancia (TIA) son ampliamente utilizados para convertir la corriente de salida de los sensores como señales de fotodiodo a tensión, la Ecuación 1 muestra la forma de calcular la tensión de salida del amplificador de transimpedancia, ya que muchos circuitos e instrumentos sólo pueden aceptar la entrada de tensión. La fig. 2.2 es un amplificador operacional con una resistencia de realimentación desde la salida a la entrada inversora es la aplicación más sencilla de tal TIA. Sin embargo, incluso este sencillo circuito de TIA requiere una cuidadosa trade-offs entre el aumento de ruido, la tensión de offset, el ancho de banda, y la estabilidad. Es evidente que la estabilidad en un TIA es esencial para un buen rendimiento, fiable.

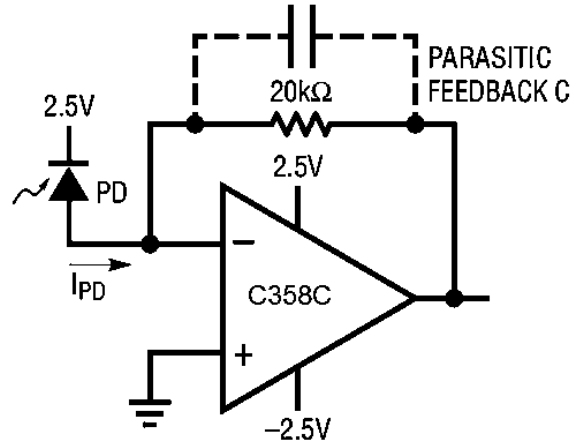


Figura 2.2 Configuración del amplificador de transimpedancia

$$V_{Out} = -I_{PD} \times R_F \dots\dots\dots \text{Ec. 1}$$

Dónde:

V_{out} = Voltaje de salida (v)

I_{PD} = Corriente del fotodiodo (mA)

R_F = Resistencia de realimentación

2.3 Fuente Simétrica

La fig. 2.3 muestra el esquema básico de una fuente de alimentación.

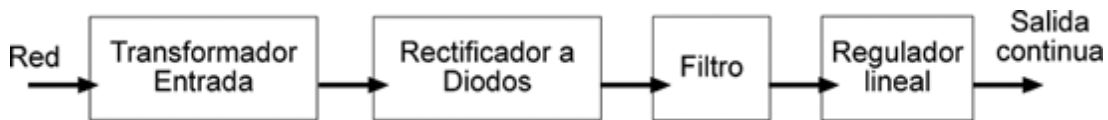


Figura 2.3 Diagrama a bloques de una fuente de voltaje

2.4 Transformador de entrada

El transformador de entrada reduce la tensión de red (fig.2.4) a otra tensión más adecuada para ser tratada. Solo es capaz de trabajar con corrientes alternas. Esto quiere decir que la tensión de entrada será alterna y la de salida también. Consta de dos arrollamientos sobre un mismo núcleo de hierro, ambos arrollamientos, primario y secundario, son completamente independientes y la energía eléctrica se transmite del primario al secundario en forma de energía magnética a través del núcleo. El esquema de un transformador simplificado es el siguiente:

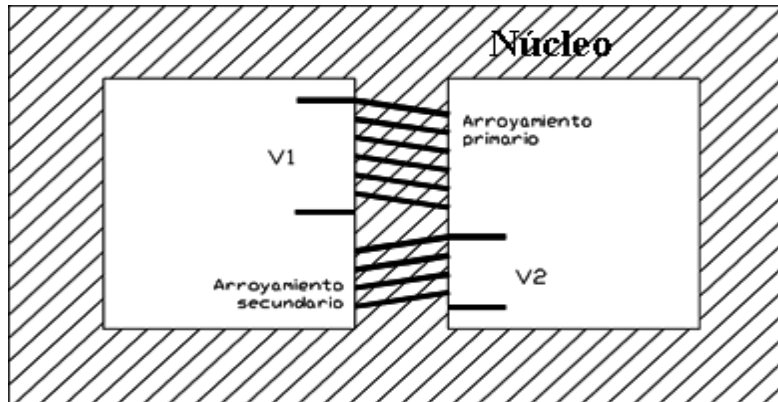


Figura 2.4 Diagrama del devanado de un transformador

La corriente que circula por el arrollamiento primario (el cual está conectado a la red) genera una circulación de corriente magnética por el núcleo del transformador. Esta corriente magnética será más fuerte cuantas más espiras (vueltas) tenga el arroyamiento primario. Si se acerca un imán a un transformador en funcionamiento se notará que el imán vibra, esto es debido a que la corriente magnética del núcleo es alterna, igual que la corriente por los arrollamientos del transformador.

En el arroyamiento secundario ocurre el proceso inverso, la corriente magnética que circula por el núcleo genera una tensión que será tanto mayor cuanto mayor sea el número de espiras del secundario y cuanto mayor sea la corriente magnética que circula por el núcleo (la cual depende del número de espiras del primario).

Por lo tanto, la tensión de salida depende de la tensión de entrada y del número de espiras de primario y secundario.

2.5 Rectificador a diodos

El rectificador es el que se encarga de convertir la tensión alterna que sale del transformador en tensión continua. Para ello se utilizan diodos. La fig.2.5 muestra la configuración de dichos diodos, un diodo conduce cuando la tensión de su ánodo es mayor que la de su cátodo.

Es como un interruptor que se abre y se cierra según la tensión de sus terminales:

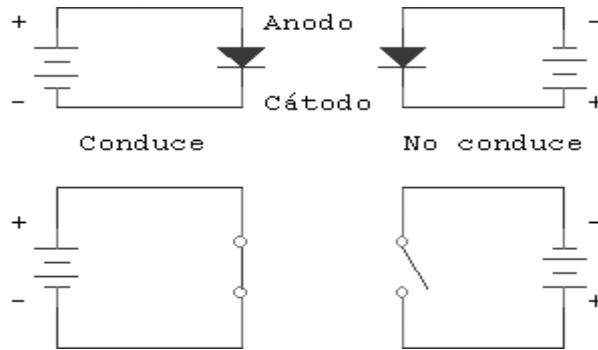


Figura 2.5 Configuración de un diodo rectificador

El rectificador se conecta después del transformador, por lo tanto le entra tensión alterna y tendrá que sacar tensión continua, es decir, un polo positivo y otro negativo en la fig. 2.6 se muestra un diagrama de conexión de un puente rectificador.

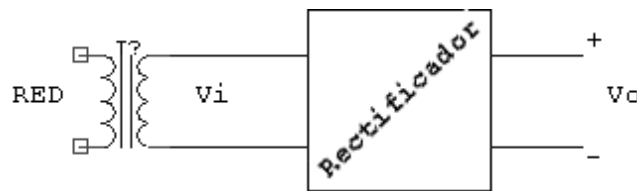


Figura 2.6 Conexión de un transformador al puente rectificador

Es tan común usar este tipo de rectificadores que se venden ya preparados los cuatro diodos en un solo componente como se muestra en la fig. 2.7. Suele ser recomendable usar estos puentes rectificadores, ocupan menos que poner los cuatro diodos y para corrientes grandes vienen ya preparados para ser montados en un radiador. Este es el aspecto de la mayoría de ellos:

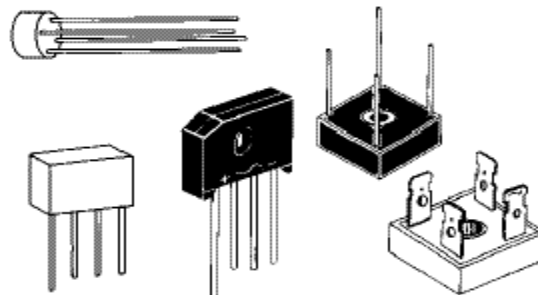


Figura 2.7 Puente de diodos

Tienen cuatro terminales, dos para la entrada en alterna del transformador, uno la salida positiva y otro la negativa o masa. Las marcas en el encapsulado suelen ser:

- ~ Para las entradas en alterna
- + Para la salida positiva
- Para la salida negativa o masa.

2.6 El filtro

La tensión en la carga que se obtiene de un rectificador es en forma de pulsos. En un ciclo de salida completo, la tensión en la carga aumenta de cero a un valor de pico, para caer después de nuevo a cero. Esta no es la clase de tensión continua que precisan la mayor parte de circuitos electrónicos. Lo que se necesita es una tensión constante, similar a la que produce una batería. Para obtener este tipo de tensión rectificadas en la carga es necesario emplear un filtro como se muestra en la fig. 2.8.

El tipo más común de filtro es el del condensador a la entrada, en la mayoría de los casos perfectamente válido. Sin embargo en algunos casos puede no ser suficiente y tiene que utilizarse componentes adicionales.

2.7 Filtro con condensador a la entrada

Este es el filtro más común, basta con añadir un condensador en paralelo con la carga (R_L), de esta forma:

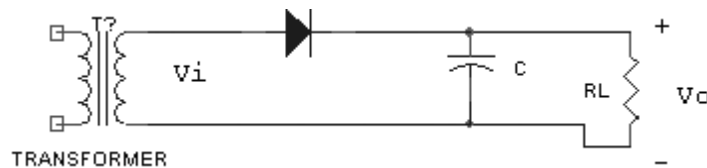


Figura 2.8 Conexión del transformador con el filtro capacitivo

2.8 El regulador

La fig. 2.9 nos muestra un regulador o estabilizador de voltaje, que es un circuito que se encarga de reducir el rizado y de proporcionar una tensión de salida de la tensión exacta que queremos.

Este es el esquema de una fuente de alimentación regulada con uno de estos reguladores:

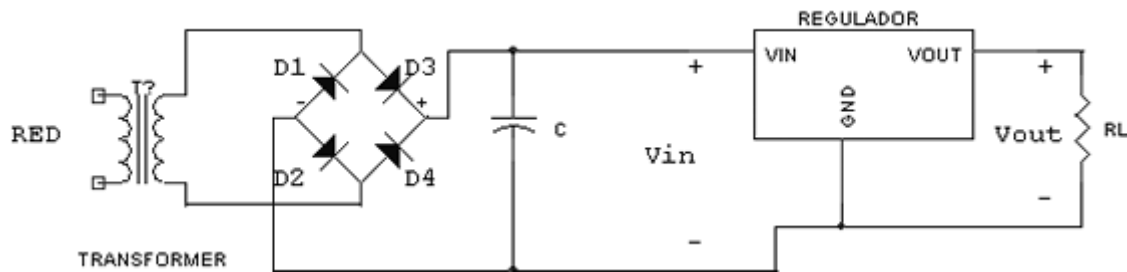


Figura 2.9 Diagrama del conexionado de un regulador de voltaje

Es muy corriente encontrarse con reguladores que reducen el rizado en 10,000 veces (80 dB), esto significa que si usas la regla del 10% el rizado de salida será del 0.001 %, es decir, inapreciable.

Las ideas básicas de funcionamiento de un regulador de este tipo son:

La tensión entre los terminales Vout y GND es de un valor fijo, no variable, que dependerá del modelo de regulador que se utilice.

La corriente que entra o sale por el terminal GND es prácticamente nula y no se tiene en cuenta para analizar el circuito de forma aproximada. Funciona simplemente como referencia para el regulador.

La tensión de entrada Vin deberá ser siempre unos 2 V o 3 V superior a la de Vout para asegurarnos el correcto funcionamiento.

En la fig. 2.10 se muestra una fuente simétrica es aquella que suministra una tensión de + X_1 voltios y otra de - X_1 voltios respecto a masa. Para ello hay que usar un transformador con doble secundario, más conocido como "transformador de toma media" o "transformador con doble devanado". En el siguiente ejemplo se ha empleado un transformador de 12 V + 12 V para obtener una salida simétrica de ± 12 V:

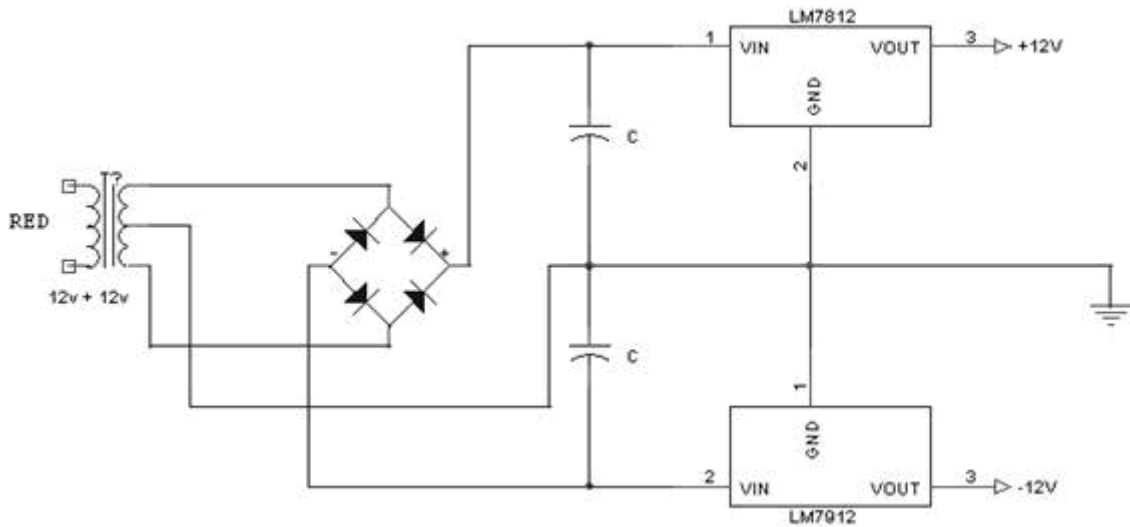


Figura 2.10 Diagrama del conexionado de una fuente de voltaje simétrica

2.9 Regulador ajustable LM317

Este regulador de tensión proporciona una tensión de salida variable sin más que añadir una resistencia y un potenciómetro. Se puede usar el mismo esquema para un regulador de la serie 78XX pero el LM317 tiene mejores características eléctricas. El aspecto es el mismo que los anteriores, pero este soporta 1.5 A. El esquema a seguir es el de la fig. 2.11.

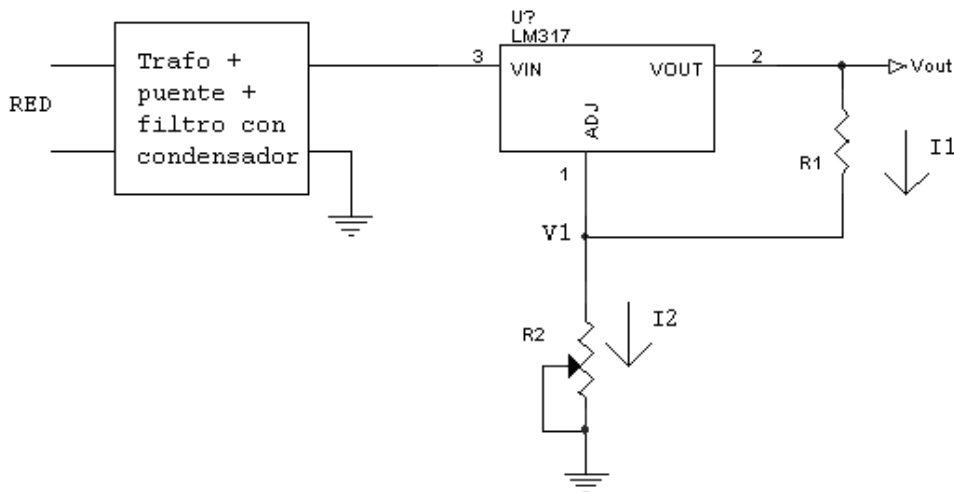


Figura 2.11 diagrama del conexionado de un regulador de voltaje

En este regulador, como es ajustable, al terminal GND se le llama ADJ, es lo mismo.

2.10 Regulador Ajustable de potencia LM350



1. Adj 2. Output 3. Input

Figura 2.12 Regulador de voltaje lm350

El LM317 es muy útil para conseguir tensiones variables, sin embargo no es capaz de suministrar más de 1.5 A, a la carga. El LM350 Fig.2.12 es otro regulador variable que funciona exactamente igual que el LM317, con la diferencia de que este es capaz por sí solo de suministrar 3 A.

2.11 Etapa de potencia

La celda de peltier y la resistencia eléctrica no pueden ser conectados directamente a las salidas del Microcontrolador pues estas salidas no tienen suficiente potencia (20 mA a 5 V) ya que estos requieren mucha potencia cada uno (7 A a 12 V), para ello se usan etapas de potencia, las cuales están formadas en su parte más importante por MOSFET, denominados MOSFET de potencia, que son capaces de conmutar rápidamente señales y grandes intensidades, la velocidad de conmutación depende en mayor medida del valor de R_{on} y del valor de la intensidad.

Por norma general suelen incluir puentes de diodos para proteger los circuitos de posibles corrientes de fuga y también unos drivers de control para mejorar y limpiar la señal, dicha señal es cuadrada modulada en anchura y se denomina PWM (Pulse Width Modulation).

Describiremos a continuación la unión de todas las partes, comunicación serie, microcontrolador...además de todas las funcionalidades de nuestra tarjeta de desarrollo y una descripción de las entradas y salidas que esta tiene.

El primer concepto que se debe tener en cuenta para controlar un motor mediante un microcontrolador es que no se pueden conectar directamente, ya que el motor tiene un consumo de corriente que el chip no le puede proporcionar. La conexión directa provocaría que se quemase la electrónica. Para evitar este desastre, se coloca una etapa de potencia entre el microcontrolador y los motores. Para que un motor gire en el sentido contrario hay que invertir la polaridad entre los bornes del mismo.

Está claro que no se puede ir detrás del robot cambiando manualmente dicha polaridad. Por lo tanto, es deseable también que la etapa de potencia nos permita cambiar el sentido de giro de los motores.

La forma más corriente de diseñar las etapas de potencia es mediante el uso de puentes. Esta configuración consiste en colocar el motor flanqueado por cuatro transistores adoptando forma de H. Veamos el funcionamiento de estos transistores.

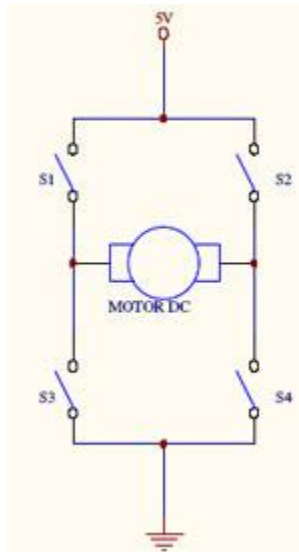


Figura 2.13 Puente en H con interruptores

Se puede pensar inicialmente que en vez de transistores, se dispone de interruptores según muestra la fig.2.13. Si se mantienen abiertos los interruptores 2 y 3, y cerrados 1 y 4, la parte derecha del motor estará conectada a tierra y la izquierda a alimentación. El motor está girando en un sentido.

Si se cierran los interruptores 2 y 3, y se abren el 1 y el 4, ahora la parte derecha del motor está conectada a alimentación, mientras que la izquierda lo está a tierra, justo al contrario que en el caso anterior. Con esta nueva configuración, el motor gira hacia el sentido contrario que antes.

Visto el funcionamiento con interruptores, extrapolar el problema a transistores es inmediato.

Se pueden emplear en el puente H tanto transistores bipolares de unión como MOSFET (transistores de efecto campo). En este caso se emplearán los segundos, ya que son capaces de soportar mayor corriente que los otros. Se usan dos tipos de MOSFET: de canal P y de canal N, tal como se muestra en la fig. 2.14.

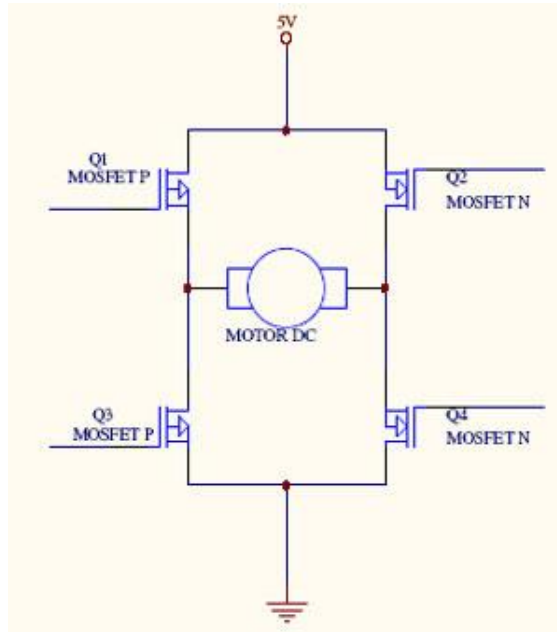


Figura 2.14 Puente en H con transistores MOSFET

Se puede pensar en un transistor MOSFET de canal P como un interruptor con señal de activación. Si la señal que le llega es un 1 el interruptor se abrirá, no dejando pasar corriente. Si la señal fuera un 0 el interruptor se cierra. Para los MOSFET de canal N sucede al contrario: se abren con un 0 y se cierran con un 1.

En el caso de las etapas de potencia del robot se podrían emplear los siguientes transistores MOSFET:

IRF9Z34N: MOSFET canal P. Soportan hasta 19 A en el drenador (en continuo). Su resistencia entre fuente y drenador es de 0.1 ohm Ω .

IRFZ24N: MOSFET canal N. Soportan hasta 14 A en el drenador (en continuo). Su resistencia entre fuente y drenador es de 0.07 ohm Ω .

2.12 Programa de apoyo

2.12.1 Matlab

MATLAB (abreviatura de *MATrix LABORatory*, "laboratorio de matrices") es una herramienta de software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M). Está disponible para las plataformas Unix, Windows, Mac OS X y GNU/Linux .

Entre sus prestaciones básicas se hallan: la manipulación de matrices, la representación de datos y funciones, la implementación de algoritmos, la creación de interfaces de usuario (GUI) y la comunicación con programas en otros lenguajes y con otros dispositivos hardware. El paquete MATLAB dispone de dos herramientas adicionales que expanden sus prestaciones, a saber, Simulink (plataforma de simulación multidominio) y GUIDE (editor de interfaces de usuario - GUI). Además, se pueden ampliar las capacidades de MATLAB con las *cajas de herramientas (toolboxes)*; y las de Simulink con los *paquetes de bloques (blocksets)*.

Es un software muy usado en universidades y centros de investigación y desarrollo. En los últimos años ha aumentado el número de prestaciones, como la de programar directamente procesadores digitales de señal o crear código VHDL. MATLAB permite crear complejos gráficos bidimensionales y tridimensionales.

En la página dedicada al estudio de los vectores vimos cómo se podía obtener la representación gráfica de un vector de datos directamente en la ventana Workspace.

En la ventana de comandos podemos obtener la misma representación proporcionando el vector de datos "y", y mandar a llamar al comando *plot* como se observa en el ejemplo 1.

Ejemplo 1:

```
>> y=[0,0.38,0.71,0.92,1.00,0.92,0.71,0.38,0];  
>> plot(y)
```

El comando *plot(x,y)* se utiliza para crear una representación gráfica bidimensional, donde (x) y (y) son vectores. Si se pasa el vector (y) de los datos, *plot (y)* MATLAB entiende que (x) es el vector de los índices (x)=[1,2,3...]; Como se observa en el ejemplo 2.

Ejemplo 2:

```
>> x=[0 10 20 30 40 50 60 70 80];  
>> y=[0 0.73 1.10 1.29 1.40 1.46 1.50 1.52 1.53];  
>> plot(x,y)
```

Esto es todo lo que se precisa para hacer representación gráfica en la ventana denominada Figure Window: Proporcionar los vectores de datos x e y (que tienen que tener el mismo número de elementos) y la llamada al comando *plot*.

Las decoraciones del gráfico (etiquetas en el eje X y en el eje Y, título, etc.) se pueden hacer mediante llamadas a comandos (xlabel, ylabel, title, etc.) o en la propia ventana gráfica, seleccionando los elementos apropiados del menú, en particular Tools->Edit Plot. (Más abajo, explicaremos brevemente esta opción)

El comando plot tiene parámetros adicionales que nos permiten especificar el color y tipo de línea o los símbolos utilizados como marcadores. Por ejemplo

```
plot(x,y,'-o','linewidth',2,'markersize',12,'markeredgecolor','g','markerfacecolor','y')
```

Los dos primeros parámetros de plot son las coordenadas de los puntos (x_i , y_i) contenidos vectores x e y. El tercer parámetro es el especificador de línea que puede contener tres datos '-mo' entre comillas simples:

Estilo de línea (- significa sólido, por defecto)

Color de línea (m significa magenta)

Marcador (o significa círculo)

En la tabla 2.1 y 2.2 se muestran las posibles opciones a utilizar.

Tabla 2.1 identificación de los símbolos y estilos a utilizar.

Color	Símbolo	Estilo de línea	Símbolo
Azul (defecto)	b	Sólido(defecto)	-
Verde	g	A puntos	:
Rojo	r	raya-punto	-.
Cian	c	rayas	--
Magenta	m		
Amarillo	y		
Negro	k		
Blanco	w		

Tabla 2.2 Opciones de símbolos para marcar puntos en la gráfica.

o	círculo	v	Triángulo (hacia abajo)
.	punto	X	Triángulo (hacia arriba)
x	marca X	<	Triángulo (hacia la izquierda)
+	más	>	Triángulo (hacia la derecha)
*	estrella	p	pentagrama
s	cuadrado	h	hexagrama
d	diamante		

Se pueden también especificar las siguientes propiedades entre comillas simples y a continuación, sus valores:

linewidth, especifica la anchura de línea su valor por defecto es 0.5

markersize, especifica el tamaño del símbolo que marca los puntos

markeredgecolor, especifica el color de borde del símbolo que marca los puntos

markerfacecolor, especifica el color de relleno del símbolo que marca los puntos

Ejemplos de *plot*:

plot(x,y), una línea azul sólida conecta los puntos (x_i, y_i)

plot(x,y,'r'), una línea roja sólida conecta los puntos (x_i, y_i)

plot(x, y,':g'), una línea de puntos de color verde conecta los puntos (x_i, y_i)

plot(x,y,'--ro','linewidth',2,'markersize',8), ejemplo más abajo

2.12.2 Vectores de datos

En la tabla 2.3 se muestra un experimento de carga de un condensador, se ha medido la diferencia de potencial V entre las placas del condensador en función del tiempo t , desde que se conecta la batería y en el ejemplo 3 podemos observar los comandos a utilizar y los resultados obtenidos se grafican en la fig.2.15.

Tabla 2.3. Resultados obtenidos de la carga de un condensador.

t (s)	0	10	20	30	40	50	60	70	80
V (V)	0.0	0.73	1.10	1.29	1.40	1.46	1.50	1.52	1.53

Ejemplo 3

```
>> t=[0 10 20 30 40 50 60 70 80];  
>> V=[0 0.73 1.10 1.29 1.40 1.46 1.50 1.52 1.53];  
>> plot(t,V,'--  
ro','linewidth',2,'markersize',6,'markeredgecolor','b','markerfacecolor'  
, 'b')  
>> title('Carga de un condensador')  
>> xlabel('t(s)')  
>> ylabel('d.d.p (V)')
```

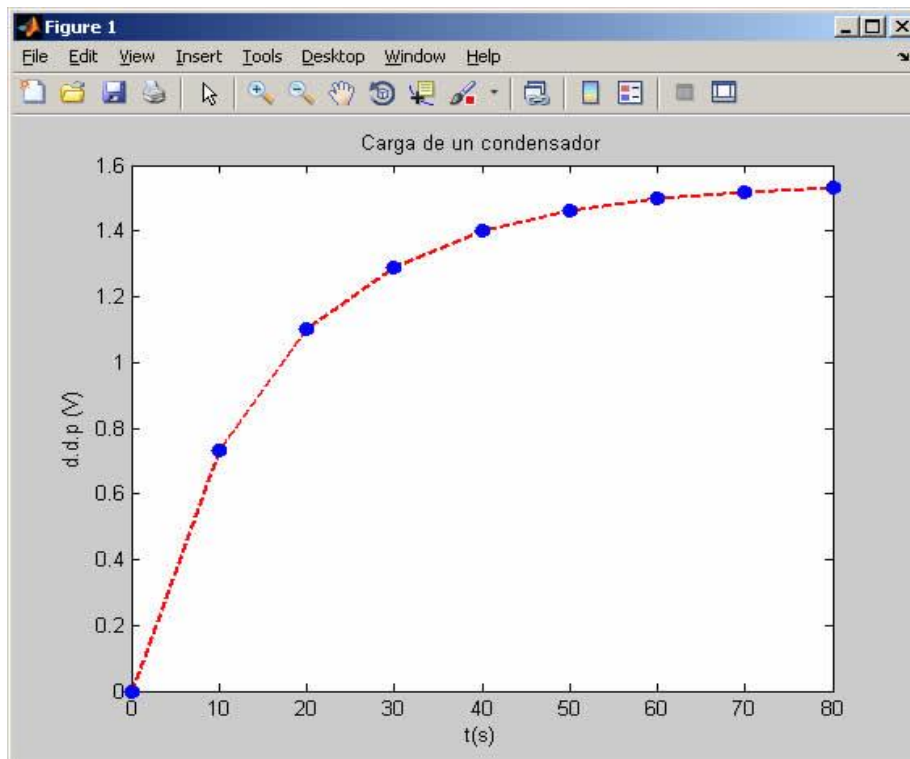


Figura 2.15 Grafica de los resultados obtenidos de la carga de un condensador

En la tabla 2.4 se muestran los resultados obtenidos de un experimento de descarga de un condensador, se ha medido la diferencia de potencial V entre las placas del condensador en función del tiempo t .

Tabla 2.4 Resultados obtenidos de la descarga de un condensador.

t (s)	10	20	30	40	50	60	70
V (V)	0.82	0.45	0.26	0.15	0.08	0.05	0.03

En el ejemplo 4 vamos a obtener una tabla de datos de la función coseno, utilizando como argumento grados, $\text{cosd}(x)$ y representarla en el intervalo de 0 a 360° , tomando un intervalo de 10° y se graficaran como se muestra en la fig. 2.16.

Ejemplo 4.

```
>> x=0:10:360;
>> y=cosd(x);
>> plot(x,y)
>> grid on %rejilla
>> title('y=cos(x)')
>> xlabel('x')
>> ylabel('y')
```

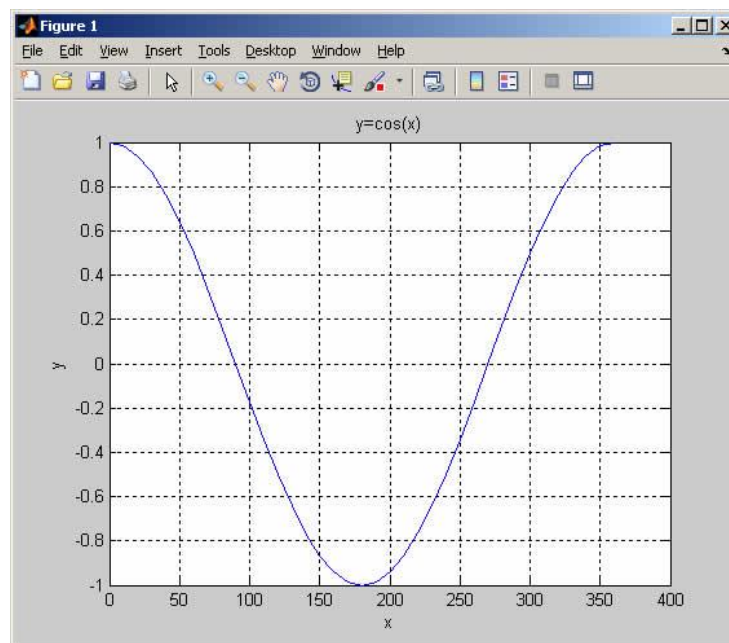


Figura 2.16 Grafica de los resultados obtenidos de la descarga de un condensador

2.12.4 Función definida en forma paramétrica

En el ejemplo 5 representaremos las denominadas figuras de Lissajous fig.2.17, que se observan en la pantalla de un osciloscopio, cuando se introducen señales senoidales de la misma o de distinta frecuencia por las entradas X e Y.

$$x = A \sin(\omega_x t)$$
$$y = A \sin(\omega_y t + \delta)$$

Ejemplo 5

```
wx=input('Frecuencia angular X: ');
wy=input('Frecuencia angular Y: ');
delta=input('Desfase (grados): ');
t=0:360;
x=sind(wx*t);
y=sind(wy*t+delta);
plot(x,y,'r');
text(0,0.2,'\omega_x/\omega_y=2/1')
xlabel('x');
ylabel('y')
title('Figuras de Lissajous')
```

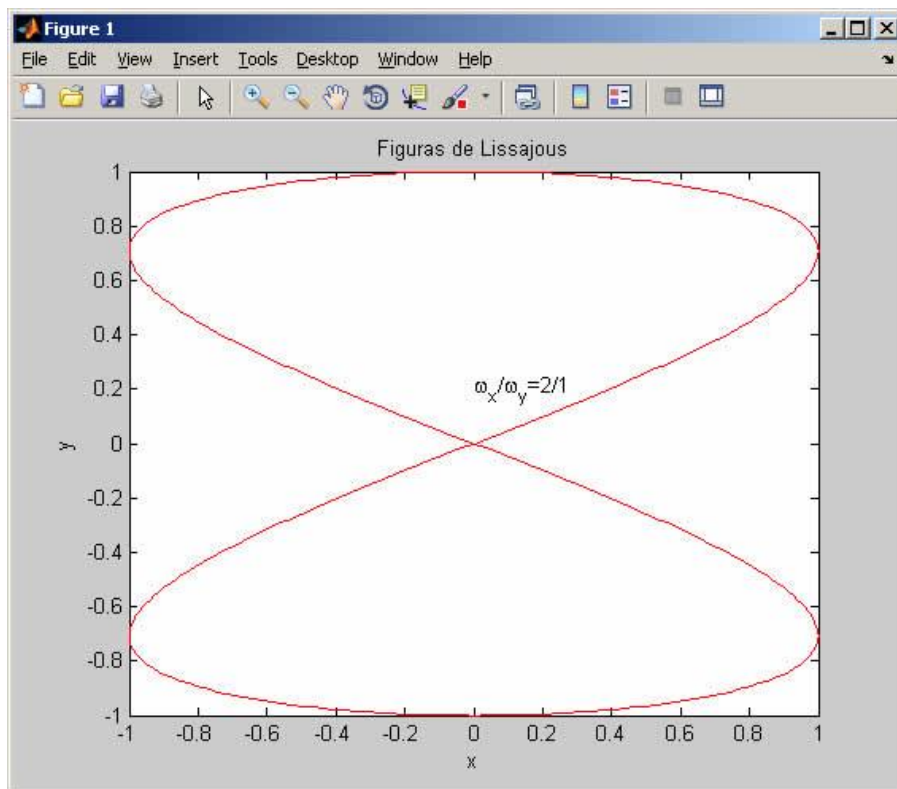


Figura 2.17 Graficas de Lissajous

Introducimos los valores de las frecuencias angulares ω_x, ω_y y del desfase δ como se muestra en el ejemplo 6.

Ejemplo 6

```
Frecuencia angular X: 2  
Frecuencia angular Y: 1  
Desfase (grados): 90
```

El comando *text* permite escribir un mensaje (caracteres entre comillas simples) en las coordenadas *x* e *y* especificadas. En este caso, el mensaje consiste en el cociente de frecuencias angulares ω_x/ω_y .

Para escribir una letra griega se antepone el carácter \. Por ejemplo, α es \alpha, ω es \omega, etc. Para poner un subíndice se antepone el carácter _. Por ejemplo para que aparezca x_1 se escribe x_1

Varias gráficas en la misma ventana, *hold*

La ecuación de una oscilación amortiguada es $x=B\exp(-\gamma \cdot t) \cdot \sin(\omega t + \delta)$ donde *B* y δ se determinan a partir de las condiciones iniciales

Sea la oscilación amortiguada $x=5.0 \cdot \exp(-7 \cdot t) \cdot \sin(100 \cdot t + 1.5)$

Queremos representar a la vez,

El desplazamiento *x* en función del tiempo *t*

La amplitud $A=5.0 \cdot \exp(-7 \cdot t)$, en función del tiempo *t*

Utilizamos el comando *hold on* para representar varias gráficas en la misma ventana. Escribimos el script *amortiguadas* Fig. 2.18 que guardamos un fichero *.M* como se muestra en el ejemplo 7.

Ejemplo 7

```
t=linspace(0,0.7,100);  
x=5*exp(-7*t).*sin(100*t+1.5);  
A=5*exp(-7*t);  
hold on  
plot(t,x,'r')  
plot(t,A,'b')  
plot(t,-A,'b')  
hold off  
legend('desplazamiento','amplitud')  
title('Oscilaciones amortiguadas')  
xlabel('t')  
ylabel('x')
```

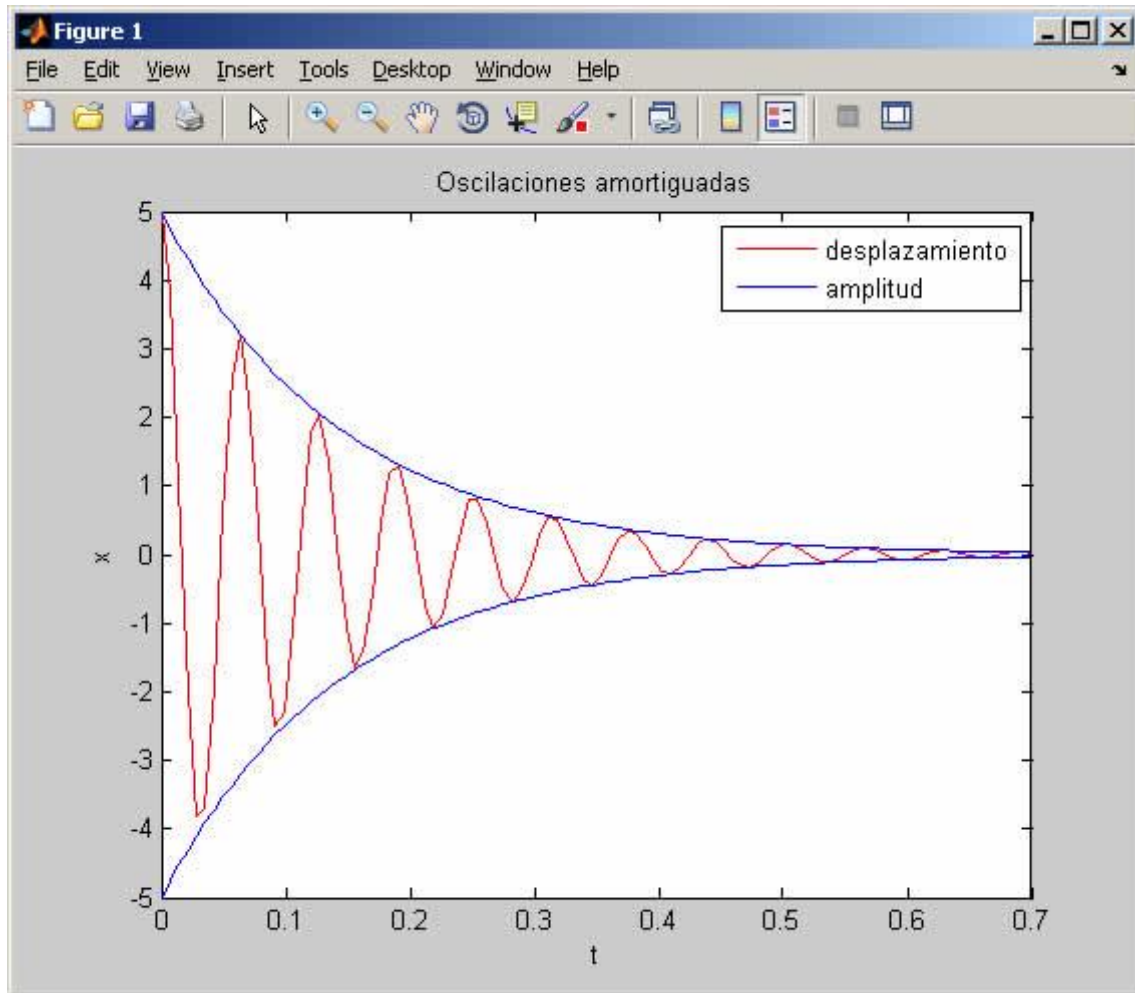


Figura 2.18 Grafica de las oscilaciones amortiguadas.

Una forma alternativa utilizando el comando *plot* se muestra en el ejemplo 8.

Ejemplo 8

```
t=linspace(0,0.7,100);
x=5*exp(-7*t).*sin(100*t+1.5);
A=5*exp(-7*t);
plot(t,x,'r',t,A,'b',t,-A) % (t,x) en rojo 'r', (t,A) en azul 'b', (t,-
A) sigue en azul
legend('desplazamiento','amplitud')
title('Oscilaciones amortiguadas')
xlabel('t')
ylabel('x')
```

O bien como se muestra en el ejemplo 9, guardando los datos de la oscilación amortiguada x , y de la amplitud A (positiva y negativa) en un vector.

Ejemplo 9

```
t=linspace(0,0.7,100);
z=[5*exp(-7*t).*sin(100*t+1.5);5*exp(-7*t);-5*exp(-7*t)];
plot(t,z)
legend('desplazamiento','amplitud')
title('Oscilaciones amortiguadas')
xlabel('t')
ylabel('x')
```

Otra forma, utilizando el comando *line* como se muestra en el ejemplo 10.

Ejemplo 10

```
t=linspace(0,0.7,100);
x=5*exp(-7*t).*sin(100*t+1.5);
A=5*exp(-7*t);
plot(t,x,'r')
line(t,A,'color','b') %propiedad: 'color', valor: 'b' (azul)
line(t,-A)
legend('desplazamiento','amplitud')
title('Oscilaciones amortiguadas')
xlabel('t')
ylabel('x')
```

Los dos primeros parámetros del comando *line* son los vectores *x* e *y* de datos. Los siguientes son propiedad entre comillas simples, seguido del valor de la propiedad, por ejemplo:

```
line(x,y,'linestyle','--','color','r','marker','o')
```

La diferencia principal entre *plot* y *line*, es que *plot* abre una ventana gráfica cada vez que es ejecutado, mientras que *line* añade gráficas a la ventana abierta previamente con *plot*, como en el script anterior.

2.12.5 Rellenando áreas de color, *fill*

En el ejemplo 11 vemos el comando *fill* que rellena áreas del color especificado como se muestra en la fig. 2.19.

Ejemplo 11

```
>> x=0:pi/40:2*pi;
>> fill(sin(x),cos(x),'g',0.7*sin(x),0.7*cos(x),'y')
>> axis equal
```

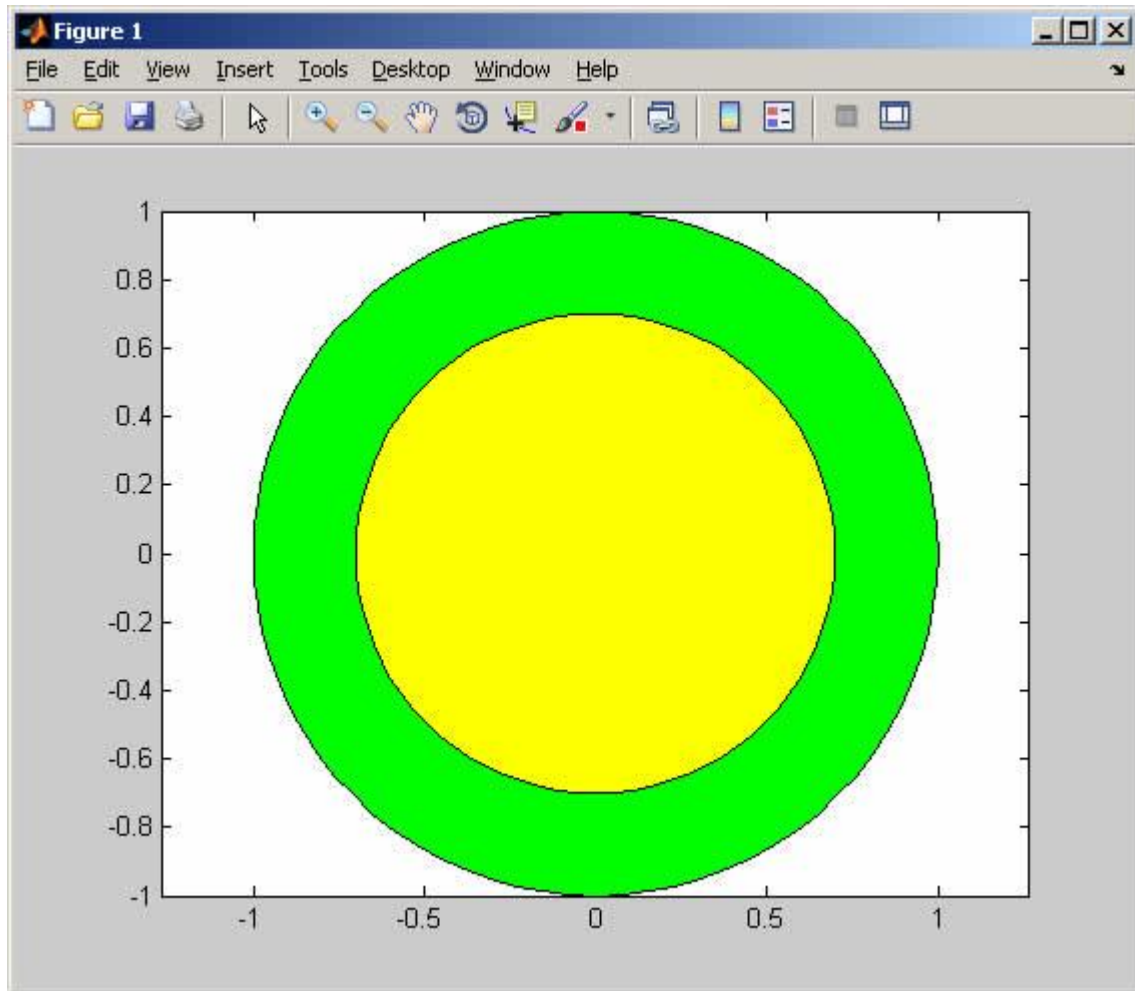


Figura 2.19 Grafica de la utilización del comando *fill*.

2.12.6 Decorando las gráficas

Ya hemos visto el efecto de *xlabel*, *ylabel*, *title* y *legend* en el ejemplo anterior para poner un título a la gráfica (*title*), identificar el eje X (*xlabel*) el eje Y (*ylabel*) y cada una de las gráficas en la misma ventana (*legend*)

2.12.6.1 legend

El comando *legend* es más complejo que los otros comandos y admite la forma siguiente:

Legend ('grafica1','grafica2',....., pos)

El orden en el que se ponen las etiquetas corresponde al que se crean las gráficas. En el ejemplo anterior, primero se crea la gráfica de la posición (t,x) y después las dos gráficas de la amplitud (t,A)

`legend('desplazamiento','amplitud')`

El último parámetro *pos* es un número opcional que especifica la posición de *legend* en la ventana. Sus posibles valores son: -1, 0, 1, 2 3, 4. El valor *pos*=1 es el valor por defecto y es el que aparece en la figura anterior. Se sugiere al lector que pruebe el efecto de los otros cinco valores.

2.12.6.2 axis

Cuando se ejecuta el comando `plot(x,y)` MATLAB crea un eje X y un eje Y cuyos límites se deducen a partir de los valores máximo y mínimo de cada uno de los vectores *x* e *y*.

En la fig. 2.20 se muestra el uso del comando *axis* que permite cambiar el rango y la apariencia de los ejes

`axis([xmin,xmax,ymin,ymax])`, establece nuevos límites a los ejes X e Y

`axis equal`, establece la misma escala para ambos ejes

`axis square`, región cuadrada

`axis tight`, ajusta los límites de los ejes al rango de los datos

En el ejemplo 12 Probamos el aspecto de una elipse de semieje mayor 2 y semieje menor 1 según los ejes.

Ejemplo 12

```
>> t=0:pi/40:2*pi;
>> plot(2*sin(t),cos(t))
>> xlabel('2·sin(t)')
>> ylabel('cos(t)')
>> axis equal
>> axis square
>> axis equal tight
```

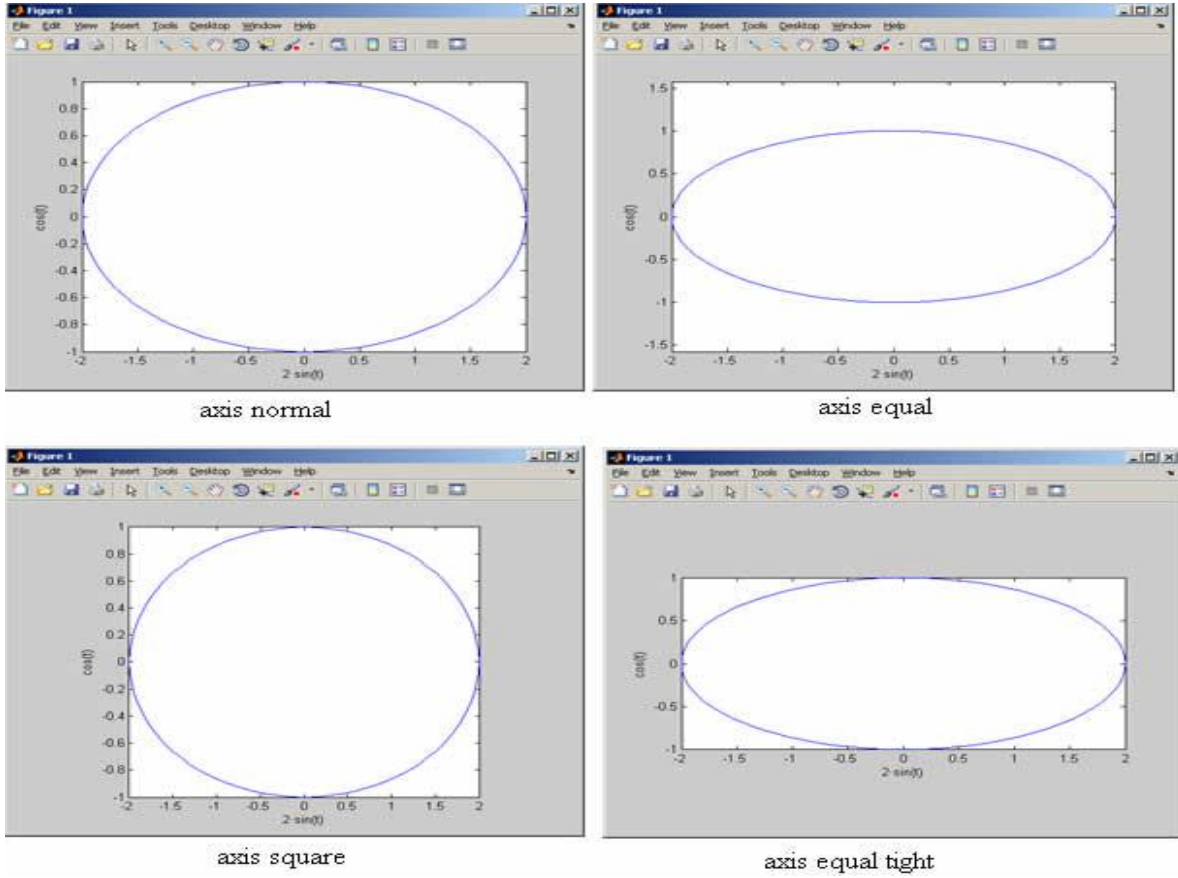



Figura 2.20 Graficas de la utilización del comando `axis`.

2.12.6.3 Autoescalado

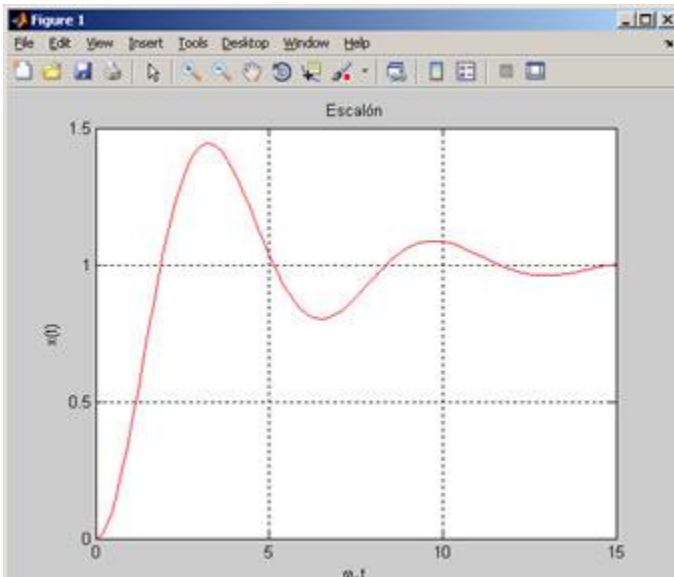


Figura 2.21 Grafica Sin el comando `axis`

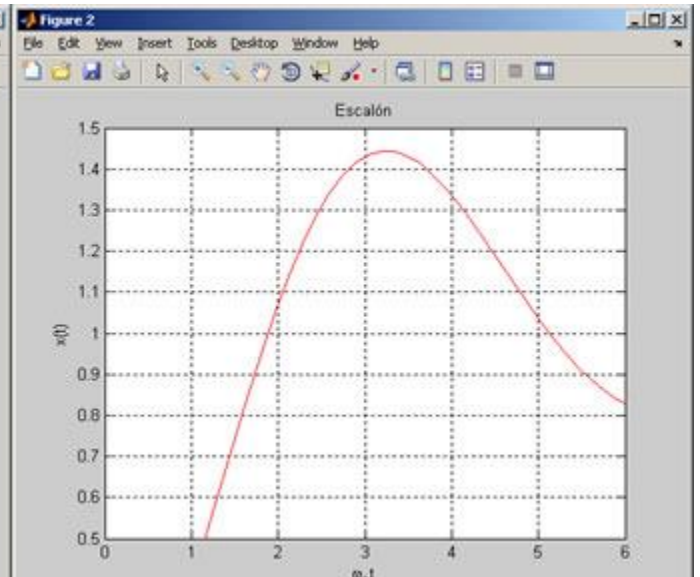


Figura 2.22 Grafica usando el comando `axis`

En la fig. 2.21 no hemos utilizado el comando *axis*, límites a los ejes, MATLAB los establece a partir de los valores máximo y mínimo de X y de Y. Sin embargo, en la fig. 2.22 hemos establecido límites a los ejes: $ymin = 0.5$ e $ymax = 1.5$, el límite $xmax = 6$, pero no hemos establecido un $xmin$ concreto. Poniendo $xmin = -Inf$ (menos infinito) dejamos a MATLAB que escale la figura. El efecto lo vemos comparando ambas figuras.

2.13 Microcontrolador

Tomando en cuenta las características de diferentes microcontroladores se seleccionó el que tiene más ventajas para su uso. Este circuito integrado forma parte de las placas Arduino UNO, que pertenece a la categoría de los microcontroladores.

Presenta características muy versátiles que le permiten aplicaciones en un gran número de variedades, además de contar con módulos Ethernet con puerto RJ-45 para la conexión a internet y la posibilidad de enviar datos. Arduino UNO. [4]

2.13.1 Arduino UNO

Arduino es una plataforma para computación física de código abierto basada en una simple tarjeta I/O y un entorno de desarrollo que implementa el procesamiento/idioma de cableado (Processing/Wiring). Arduino puede ser usado para desarrollar objetos autónomos interactivos o puede ser conectado a software en su ordenador (por ejemplo, Flash, Processing, MaxMSP), en la tabla 2.5 se muestran las características eléctricas.

El IDE de código abierto puede ser descargado de forma gratuita (en la actualidad para Mac OS X, Windows y Linux).

Tabla 2.5 Características eléctricas del Arduino UNO

Microcontroladores	ATmega328P
Tensión de funcionamiento	5 V
Voltaje de entrada (recomendado)	7-12 V
Voltaje de entrada (límite)	6-20 V
Digital pines I / O	14 (de las cuales 6 proporcionan salida PWM)
PWM digital pines I / O	6
Pines de entrada analógica	6
Corriente DC por E / S Pin	20 mA
Corriente DC de 3.3V Pin	50 mA
Memoria flash	32 KB (ATmega328P) de los cuales 0,5 KB utilizado por el gestor de arranque
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Velocidad de reloj	16 MHz
Largo	68,6 mm
Anchura	53.4 mm
Peso	25 g

A continuación se muestra en la fig.2.23 donde están ubicados los elementos más importantes que componen la placa Arduino UNO que son descritos de arriba abajo y de izquierda a derecha:

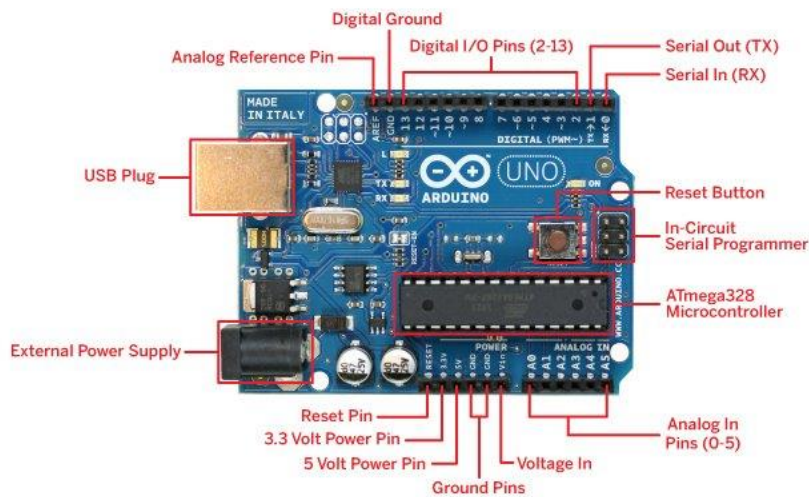


Fig. 2.23 Estructura del Arduino UNO

2.13.2 Estructura básica de un programa

La estructura básica de programación de Arduino es bastante simple y divide la ejecución en dos partes:

Setup

Loop

Setup() constituye la preparación del programa y loop() es la ejecución. En la función Setup() se incluye la declaración de variables y se trata de la primera función que se ejecuta en el programa. Esta función se ejecuta una única vez y es empleada para configurar las terminales de salida mediante pinMode(). La función loop() incluye el código a ser ejecutado continuamente (leyendo las entradas de la placa, salidas, etc.).

```
void setup()
{
  //Preparacion del Programa
}

void loop()
{
  //Ejecución
}
```

Nota: Al igual que en C se pueden introducir bloques de comentarios con /*...*/.

Entradas/Salidas digitales y analógicas.

digitalRead(pin). Lee el valor desde un pin digital específico. Devuelve un valor HIGH o LOW.

digitalWrite(pin, value). Introduce un nivel alto (HIGH) o bajo (LOW) en el pin digital especificado. De nuevo, el pin puede ser especificado con una variable o una constante.

analogRead(pin). Lee el valor desde el pin analógico especificado con una resolución de 10 bits. Esta función solo funciona en los pines analógicos (0-5). El valor resultante es un entero de 0 a 1023. Los pines analógicos, a diferencia de los digitales no necesitan declararse previamente como INPUT o OUTPUT.

analogWrite(pin, value). Escribe un valor pseudo-analógico usando modulación por ancho de pulso (PWM) en un pin de salida marcado como PWM.

2.13.3 Tipos de Datos

Arduino permite manejar los siguientes tipos de datos:

Byte. Almacena un valor numérico de 8 bits. Tienen un rango de 0-255.

Int. Almacena un valor entero de 16 bits con un rango de 32,767 a -32,768.

Long. Valor entero almacenado en 32 bits con un rango de 2, 147, 483,647 a -2, 147,483, 648.

Float. Tipo coma flotante almacenado en 32 bits con un rango de 3.4028235E+38 a -3.4028235E+38.

Arrays. Se trata de una colección de valores que pueden ser accedidos con un número de índice (el primer valor del índice es 0).

2.13.4 Funciones de Tiempo

delay(ms). Realiza una pausa en el programa la cantidad de tiempo en milisegundos especificada en el parámetro (máximo 1000, mínimo 1).

millis(). Devuelve la cantidad de milisegundos que lleva la placa Arduino ejecutando el programa actual como un valor long unsigned. Después de 9 horas el contador vuelve a 0. [1]

2.14 Sensor DS18B20

Este es el último sensor DS18B20 con salida digital de un solo hilo de la compañía Maxim IC. Reporta grados Celsius con una precisión de 9-12 bits, -55 °C a 125 °C (+/-0.5 °C). Cada sensor tiene un único número serial de 64 bits grabado internamente, lo que posibilita que una gran cantidad de sensores sean conectados al mismo bus, en la fig.2.24 se muestra el diagrama de conexión hacia el microcontrolador.

Características:

- Interfaz de 1 solo hilo (1-Wire®) que requiere un solo pin para comunicaciones
- Cada dispositivo cuenta con un único número de serie de 64 bits almacenado en su ROM interna
- Capacidad de multidespliegue (Multidrop) simplifica las aplicaciones de monitoreo distribuido de temperatura
- No requiere componentes externos
- Puede ser energizado desde la línea de datos. Rango de voltaje de alimentación: 3.0 V a 5.5 V
- Mide la temperatura de -55 °C a +125 °C (-67 °F a +257 °F)
- Precisión de ±0.5 °C de -10 °C a +85 °C
- Resolución seleccionable por el usuario de 9 a 12 bits
- Convierte la temperatura a una palabra de 12 bits en 750 ms (máx.)
- Configuraciones de alarma definibles por el usuario no volátiles (NV)

- Comando de búsqueda de alarma identifica la dirección del dispositivo cuya temperatura está fuera de los límites pre-programados (condición de alarma de temperatura)
- Las aplicaciones incluyen controles de termostato, sistemas industriales, productos de consumidor, termómetros o cualquier sistema termalmente sensible

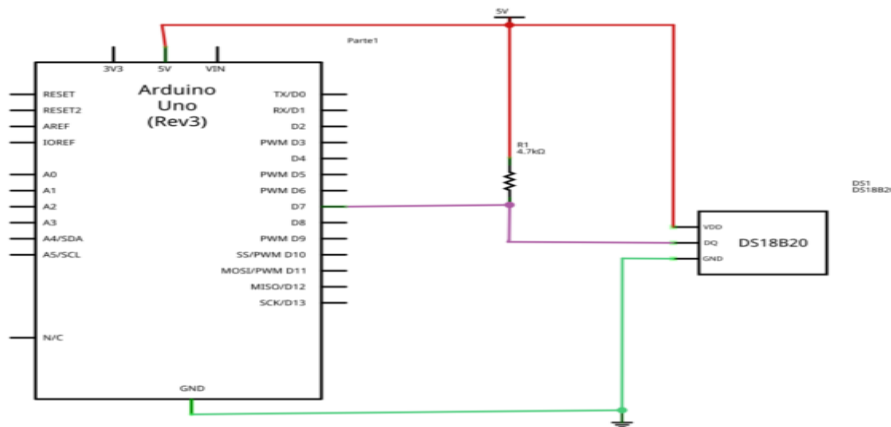


Fig. 2.24 Diagrama del sensor DS18B20

2.15 Laser

Un láser (de la sigla inglesa *Light Amplification by Stimulated Emission of Radiation*, amplificación de luz por emisión estimulada de radiación) es un dispositivo que utiliza un efecto de la mecánica cuántica, la emisión inducida o estimulada, para generar un haz de luz coherente tanto espacial como temporalmente. La coherencia espacial se corresponde con la capacidad de un haz para permanecer con un pequeño tamaño al transmitirse por el vacío en largas distancias y la coherencia temporal se relaciona con la capacidad para concentrar la emisión en un rango espectral muy estrecho, en la fig. 2.25 se muestra el láser OPF372 que se utilizara en el proyecto.



Fig. 2.25 Laser OPF372A Emisor

2.16 Fotodetector

Un fotodetector es un sensor que genera una señal eléctrica dependiente de la luz u otra radiación electromagnética que recibe fig. 2.26. Algunos están basados en el efecto fotoeléctrico, otros en el fotovoltaico, otros en el fotoelectroquímico y otros en la fotoconductividad.



Fig. 2.26 Fotodetector OPF472 Receptor

2.17 Fibra multimodo

La fibra óptica multimodo es un tipo de fibra óptica mayormente utilizada en el ámbito de la comunicación en distancias cortas, como por ejemplo un edificio o un campus. Los enlaces multimodo típicos tienen un ratio de datos desde los 10 Mbit/s a los 10 Gbit/s en distancias de hasta 600 metros (2000 pies) --- más que suficiente para cumplir las premisas de distintas aplicaciones.

2.17.1 Aplicaciones

El equipo utilizado para las comunicaciones sobre fibra óptica multimodo es más barato que el utilizado para la fibra óptica monomodo. La velocidad típica de transmisión los límites de distancia está ubicado en los 100 Mbit/s en distancias de hasta 2 kilómetros (100BASE-FX), 1 Gbit/s hasta 1000 metros y 10 Gbit/s hasta los 550 metros.

Dada su alta capacidad y fiabilidad, la fibra óptica multimodo se utiliza generalmente para aplicaciones troncales en edificios. Un número de usuarios en aumento disfrutan de los beneficios de la fibra cercana al usuario utilizándola en su propio domicilio.

Las arquitecturas estándar como los cableados centralizados ofertan al usuario la habilidad de hacer uso de las capacidades de la fibra en cuanto a distancia centralizando la electrónica en una habitación dedicada a ésta.

2.17.2 Tipos

Las fibras multimodo son diferenciadas basándose en su núcleo y el diámetro de su revestimiento. Así, un cable de fibra óptica 62.5/125 μm tiene un núcleo del tamaño de 62.5 micrómetros y un diámetro de 125 micrómetros de revestimiento véase la tabla 2.2. La transición entre el núcleo y el revestimiento puede ser estricta o gradual. Estos dos tipos poseen características de transición distinta y también diferente distancia de propagación eficiente.

En adición a lo ya explicado, las fibras multimodo también se identifican por un sistema de clasificación determinado por el estándar ISO 11801 --- OM1, OM2 y OM3 --- el cual está basado en el ancho de banda modal de la fibra multimodo.

El OM4 se finalizó en agosto de 2009, y fue publicado al final del mismo año. El cableado OM4 soportaría conexiones de 125 metros entre 40 y 100 Gbit/s. Las letras OM significan "Optical Multi-mode".

Durante muchos años, el OM1 y OM2 tuvieron muy diversas aplicaciones. Estas fibras soportan fácilmente aplicaciones desde Ethernet (10 Mbit/s) hasta GbEthernet (10 Gbit/s) y fueron ideales para su utilización junto con transmisores LED.

Las fibras con designación OM3 proveen el ancho de banda suficiente para soportar conexiones Ethernet de 10 Gbit/s hasta los 300 metros. Los fabricantes de fibra óptica han refinado enormemente su proceso de manufacturación desde que el estándar fue implantado y los cables fueron hechos para soportar 10 GbE hasta 400 metros. La migración hasta LOMMF/OM3 ha hecho que los usuarios migren a conexiones más rápidas.

Los LEDs tienen una tasa de modulación máxima de 622 Mbit/s ya que no pueden ser encendidos o apagados lo suficientemente rápido para soportar aplicaciones que requieran un ancho de banda mayor. Los VCSELs soportan una modulación de hasta 10 Gbit/s y son utilizados en muchas redes de alta velocidad.

Los cables se distinguen por el color del revestimiento que los cubren: para el OM1 y OM2 se recomienda el color naranja, mientras que el color "Aqua" es el recomendado para los OM3 y OM4, para definir que están optimizados para láser.

La dispersión modal de los VCSEL es medida por el "differential modal delay" (DMD). Ésta es causada por diferentes velocidades en modos individuales de un pulso de luz. El efecto red causa que la luz se esparza conforme va aumentando la distancia. Cuando mayor sea la distancia, mayor será la dispersión.

Tabla 2.6 Tabla Comparativa Fibra Óptica multimodo y monomodo

Categoría	Ancho de banda modal mínimo	100 Mb Ethernet 100BASE-FX	1 GB (1000 Mb) Ethernet 1000BASE-SX	10 GB Ethernet 10GBASE-SR	40 GB Ethernet	100 GB Ethernet
OM1 (62.5/125)	200 / 500 MHz.km	Hasta 2000 metros (FX)	275 metros (SX)	33 metros (SR) _i	No soportado	No soportado
OM2 (50/125)	500 / - MHz.km	Hasta 2000 metros (FX)	550 metros (SX)	82 metros (SR) _i	No soportado	No soportado
OM3 (50/125) Laser Optimized	1500 / 2000 MHz.km	Hasta 2000 metros (FX)	550 metros (SX)	300 metros (SR) _i	100 metros 330 metros QSFP+ eSR4	100 metros
OM4 (50/125) Laser Optimized	3500 / 4700 MHz.km	Hasta 2000 metros (FX)	1000 metros (SX)	400 metros (SR) _i	150 metros 550 metros QSFP+ eSR4	150 metros

2.18 Conectores de Fibra multimodo (ST)

ST (una marca registrada de AT&T) es probablemente todavía el conector más popular para las redes multimodo (fig.2.27), instalado en la mayoría de los edificios y campus. Tiene una montadura de bayoneta y una férula larga y cilíndrica de 2.5 mm usualmente de cerámica o polímero para sostener a la fibra. La mayoría de las férulas son de cerámica pero hay algunas de ellas de metal o plástico. Y debido a que tienen un resorte interno, se debe asegurar que se insertan adecuadamente. Si tiene pérdidas altas, vuelva a conectarlos para ver si se tiene una mejor conexión y menor pérdida. Los conectores ST/SC/FC/FDDI/ESCON tienen el mismo diámetro de la férula de - 2.5 mm o aproximadamente 0.1 de pulgada –por lo que pueden ser mezclados y acoplados utilizando adaptadores de acoplamiento híbridos. Esto permite realizar pruebas en forma muy conveniente y sencilla, debido a que se puede tener un solo juego de cables de referencia multimodo con conectores ST o SC y los adaptadores para todos estos conectores.



Fig. 2.27. Conectores ST

CAPÍTULO 3. DESARROLLO E IMPLEMENTACIÓN DEL PROYECTO

3.1 Diseño del sistema de adquisición de datos

3.1.1 Sensor utilizado, DS18B20

En la fig.3.1. Se muestra la fotografía de un sensor de temperatura real y su diagrama de conexión, el cual ya explicamos en el capítulo anterior, aquí vamos a ver su aplicación en el desarrollo del proyecto.



Fig. 3.1 Sensor DS18B20

3.1.2 Celda de peltier

Los enfriadores termoeléctricos (TEC o Peltier) crean un diferencial de temperatura en cada una de sus caras. Una de ellas se calienta mientras la otra se enfría. Por lo tanto podrás utilizarlo para calentar o enfriar algo dependiendo del lado que emplees.

La placa Peltier trabaja muy bien mientras realices una disipación correcta... Después de encender este dispositivo el lado "caliente" se calentará rápidamente y el otro lado se enfriará también en un corto tiempo. Si no eliminas el calor del lado caliente mediante un disipador el Peltier entrará en equilibrio y por lo tanto no hará nada (ni enfriará ni calentará) en la fig.3.2 se muestra la imagen de una celda de peltier.



Fig. 3.2 Celda de peltier

3.1.3 Diagrama a bloques

En la fig.3.3. Se muestra un diagrama a bloques donde veremos el funcionamiento del Sensor de temperatura, con todos los componentes que lo integran, así como la interfaz gráfica visualizada en la computadora y al mismo tiempo guardar los datos obtenidos en una hoja de datos Excel.

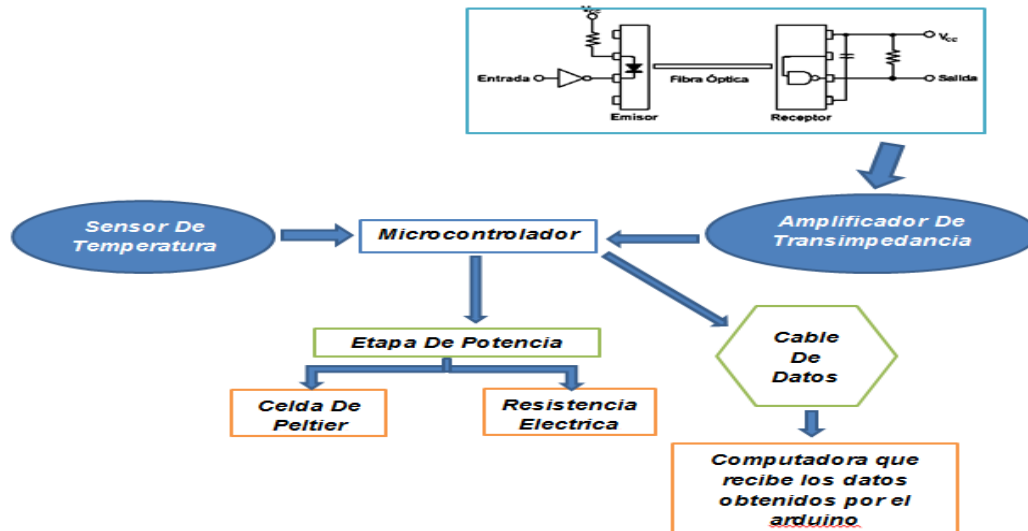


Fig. 3.3 Diagrama a bloques de funcionamiento.

3.2 Construcción de la etapa de potencia

3.2.1 Celda de peltier y Resistencia Eléctrica

Para el control de la celda de peltier y la resistencia eléctrica se utilizó el circuito de la Fig. 3.4, para poder controlarlo con el Arduino mediante una señal PWM de 5 volts. La programación de esta se basa mediante una comparación de mediciones, las temperaturas que se utilizaron son de 11 °C a 90 °C, al iniciar el sistema se enciende la celda de peltier hasta llegar a los 11 °C una vez alcanzada esa temperatura la celda se apaga y se enciende la resistencia eléctrica y se empieza a tomar los datos en MATLAB hasta llegar a los 90 °C.

El circuito del control de potencia para la celda de peltier y la resistencia eléctrica se muestra en la Fig. 3.4.

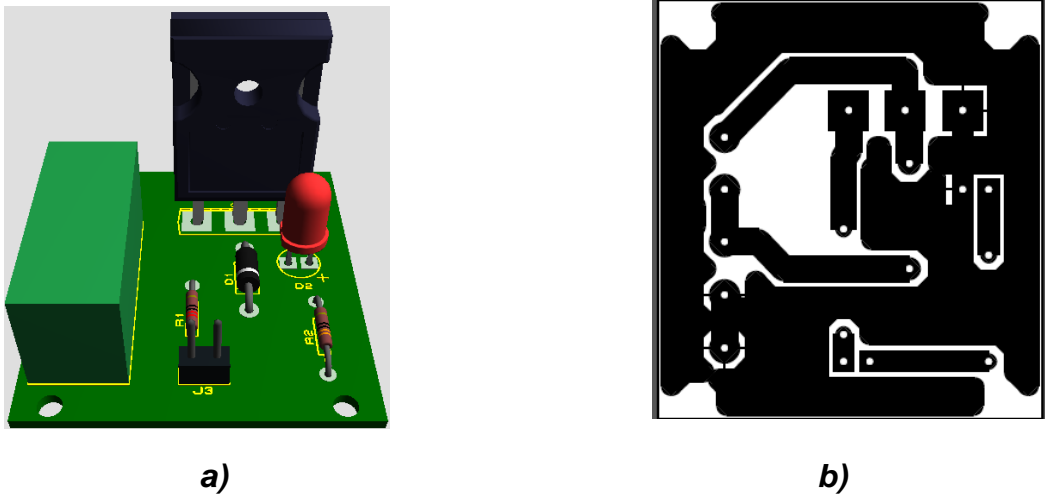


Fig. 3.4 Circuito de potencia para controlar la celda de peltier y la resistencia eléctrica, a) Prototipo; b) PCB, diseñado en ISSI PROTEUS.

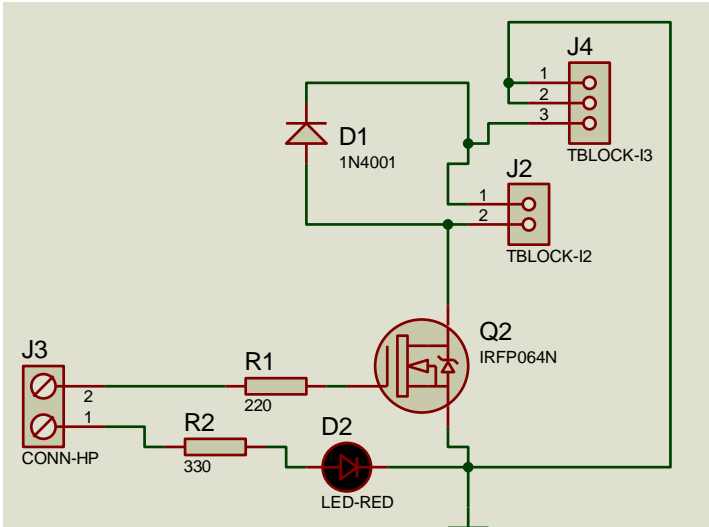


Fig. 3.5 Diagrama eléctrico del circuito de potencia.

3.3 Interface gráfica de MATLAB

A continuación se muestra en la fig.3.8 la interfaz graficas de usuario de MATLAB, en el cual se podrá ver los valores obtenidos de los sensores en tiempo real.

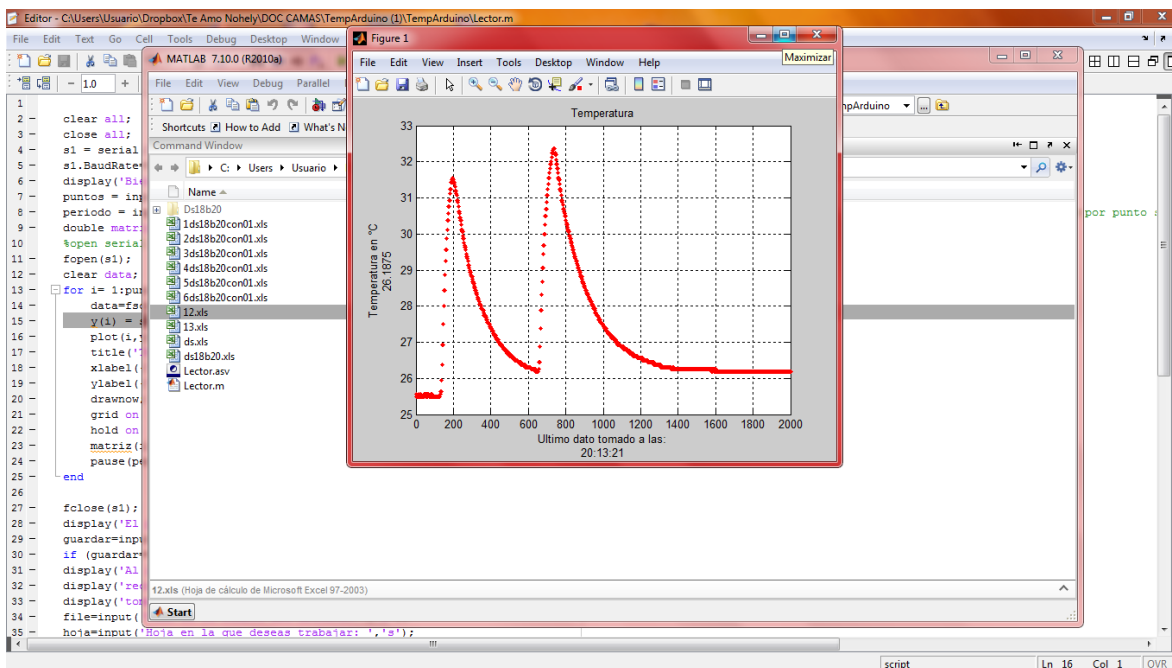


Fig. 3.8 Interfaz de usuario MATLAB.

CAPÍTULO 4. PRUEBAS Y RESULTADOS

4.1 Resistencia Eléctrica

En la fig.4.1 se muestra la gráfica de las primeras pruebas que se hicieron con la resistencia eléctrica, tomando en cuenta que la temperatura máxima a alcanzar es de 90 °C pusimos a calentar la resistencia para ver en qué tiempo alcanzaba la temperatura máxima y saber también si llegaría a la temperatura deseada que son los 90 °C, las primeras pruebas arrojaron resultados no tan deseado por que alcanzó los 50 °C en 2 horas así que tuvimos que poner otra resistencia en paralelo para lograr alcanzar mayor temperatura en menor tiempo, al hacer las pruebas nuevamente con la nueva resistencia integrada el tiempo que llevo para alcanzar los 70 °C es de 40 minutos, es un buen tiempo pero al aumentar una resistencia en paralelo la corriente que demanda es mayor y se calienta el circuito de potencia hecho por eso se optó por dejar la temperatura máxima como 70 °C.

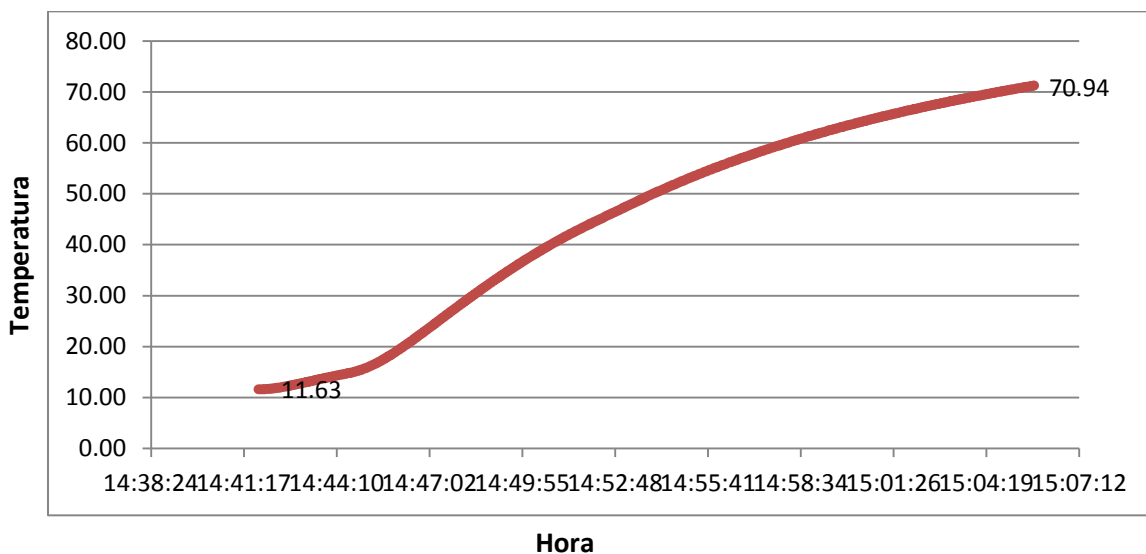


Fig. 4.1 Grafica de la temperatura de la resistencia eléctrica.

4.2 Celda de peltier

Las pruebas con la celda de peltier fueron totalmente distintas a las obtenidas con la resistencia eléctrica ya que en la primera prueba obtuvimos una temperatura de 11 °C la cual es menor a la temperatura mínima deseada que es de 20 °C Fig.4.2, así que para el proceso de caracterización se emplearan las temperaturas de 11 °C a 70 °C.

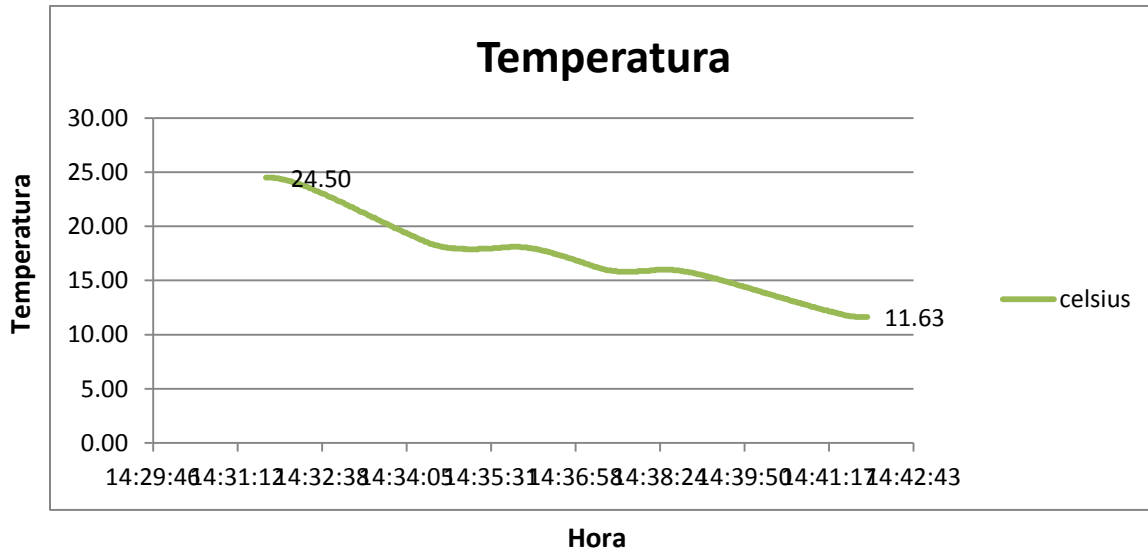


Fig. 4.2 Grafica de la temperatura de la celda de peltier.

4.3 Amplificador de transimpedancia

Esta etapa del proyecto fue la más problemática puesto a que es complicado conseguir este tipo de circuitos, así que optamos por diseñar nuestro amplificador de transimpedancia, las primera pruebas se muestran en la fig.4.3 al analizar la muestra me di cuenta que los resultados fueron erróneos ya que el OP AMP que utilizamos necesita una fuente simétrica por la que tuvimos que diseñarla también, otro problema que tuvimos fue el de tener que regular la ganancia en el OP AMP ya que el láser nos proporciona apenas $29\mu W$ que es una potencia bastante pequeña así que usamos un potenciómetro para poder controlar la ganancia.

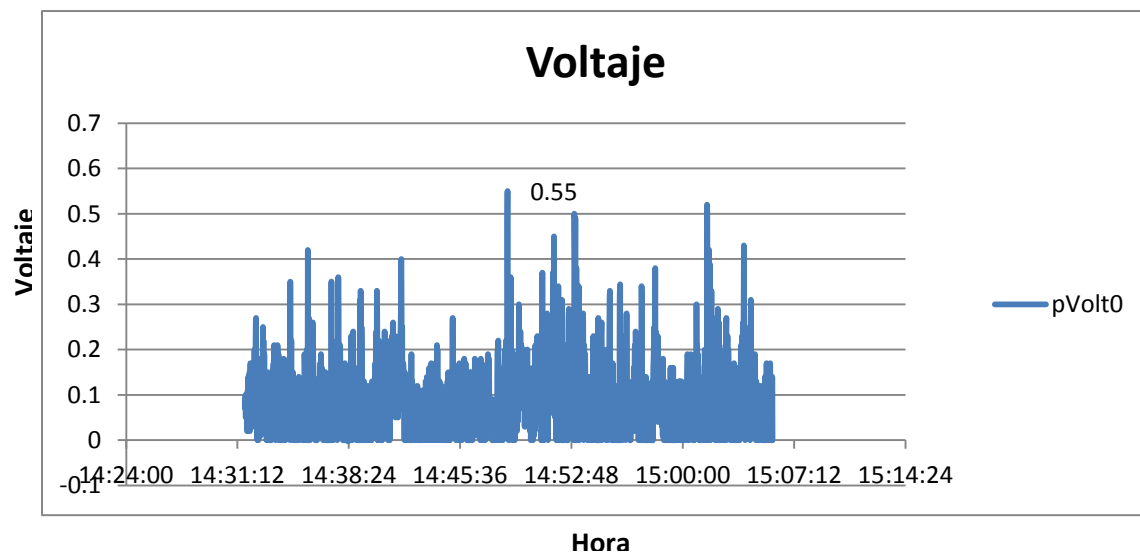


Fig. 4.3 Grafica del sensor de transimpedancia.

4.4 Funcionamiento

Una vez teniendo todo listo y probado llego la hora de hacer las pruebas finales en este cargamos el código en C al microcontrolador y fijamos la temperatura mínima y máxima a alcanzar por el sistema, colocamos la fibra óptica dentro del cubo de vidrio y empezamos a enfriar la fibra con la celda de peltier hasta llegar a los 11 °C en la fig. 4.4 se muestra la gráfica de la primer prueba realizada, cuando la temperatura llega a los 11 °C la celda de peltier se apaga y se encienden las resistencias para calentar la fibra hasta llegar a los 70 °C, durante ese proceso de enfriamiento y calentamiento Matlab monitorea lo que pasa con el sensor de transimpedancia y el sensor de temperatura y grafica la temperatura y guarda los datos obtenidos en una hoja de datos de Excel para que podamos ver donde hubo una variación de voltaje y así caracterizar el sensor.

En la fig. 4.5 y 4.6 se muestran los resultados de las pruebas hechas en 2 días distintos en ellas se pueden observar que existe muchas variaciones en cuanto a los resultados.

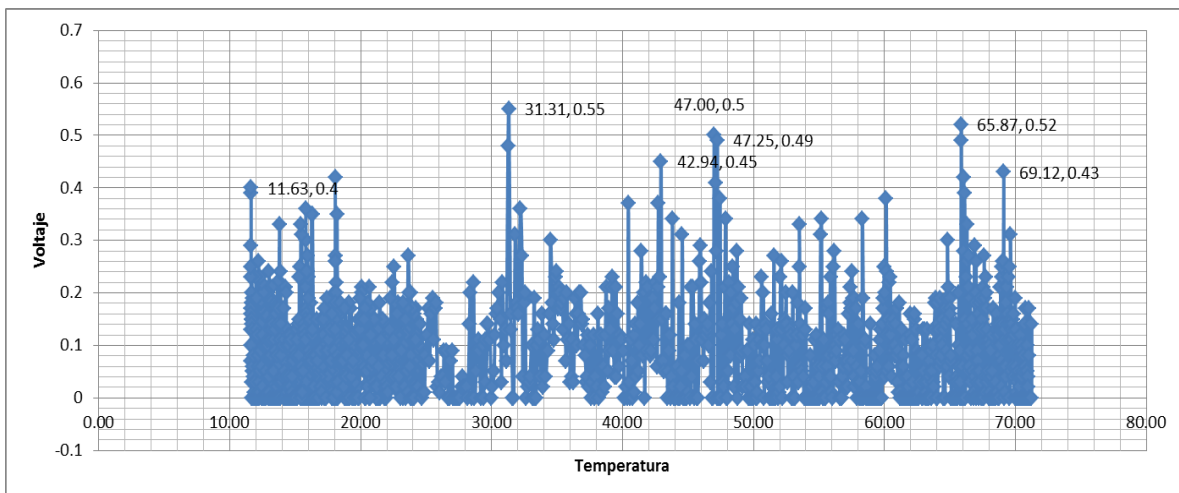


Fig. 4.4 Grafica de la temperatura y el voltaje del sensor de transimpedancia prueba 1.

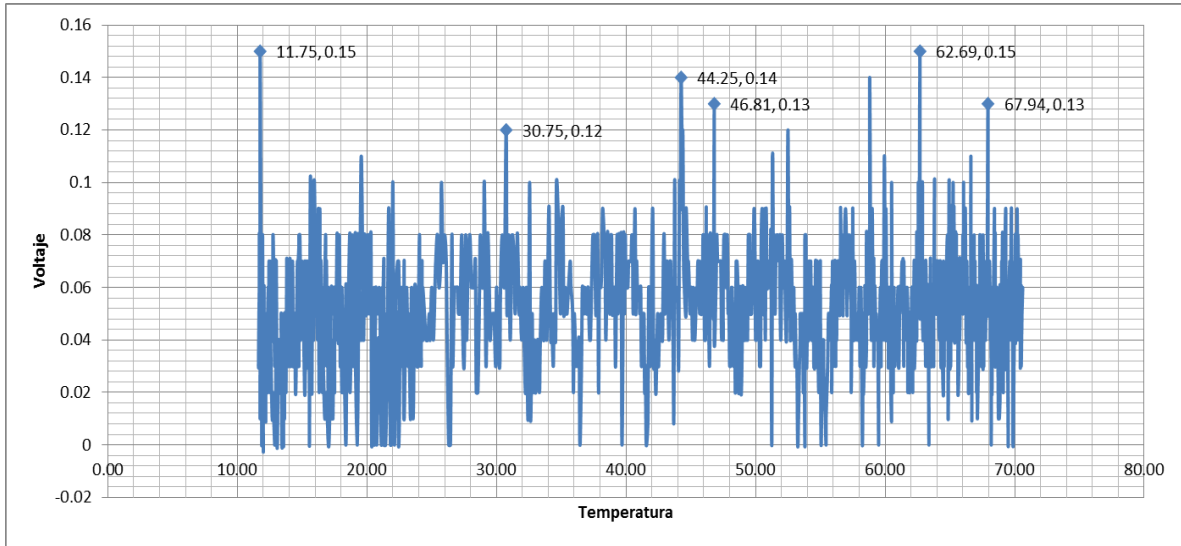


Fig. 4.5 Grafica de la temperatura y el voltaje del sensor de transimpedancia prueba 2.

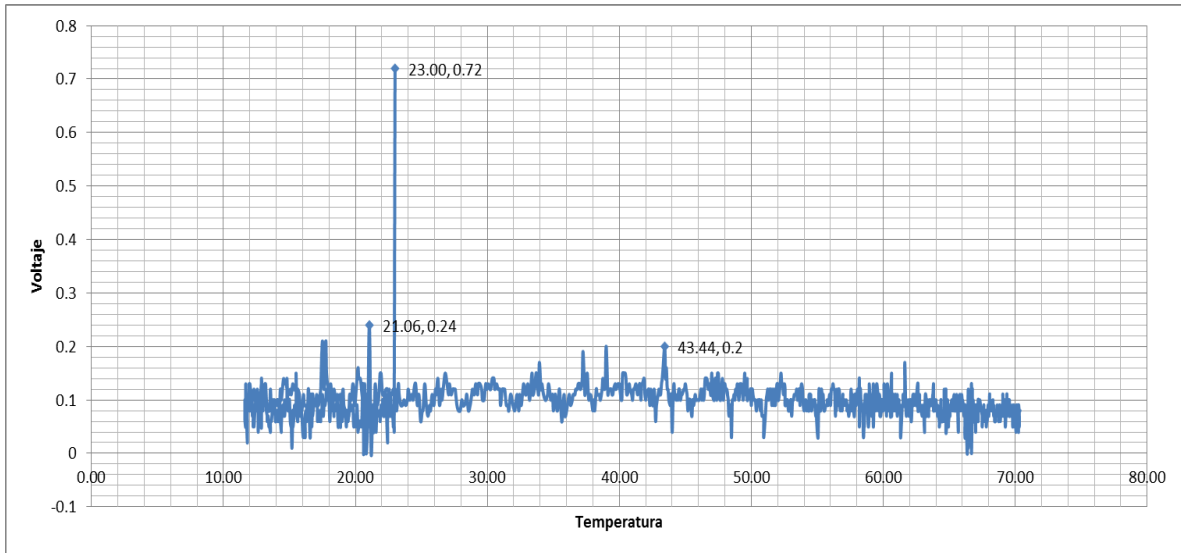


Fig. 4.6 Grafica de la temperatura y el voltaje del sensor de transimpedancia prueba 3.

Conclusiones

En conclusión podemos decir que al llevar a cabo este proyecto nos dimos cuenta de varios factores que afectan el funcionamiento de un sensor de temperatura basado en fibra óptica una de ellas es la temperatura que con cualquier leve incremento de temperatura esta reacciona por eso se realizó este proyecto al hacer la pruebas y la caracterización podemos observar que hay muchas variaciones en cuanto a la salida del OP AMP esto se debe al constante cambio de temperatura que hay en el medio (cubo de cristal) o por el la variación del voltaje que alimenta el láser de cualquier forma es otro de los tantos factores que afectan en el funcionamiento del sensor(F.O), y ya para finalizar se logró el objetivo planteado descrito al inicio de este documento mis sensaciones son satisfactorias ya que en ciertas etapas llegue a pensar que esto no funcionaría por diversos factores que se dieron durante el proceso.

Referencias

- [1] Udd, E., Editor, "Fiber optic sensor: an introduction for Engineers and Scientists" , John Wiley and Sons, 1991.
- [2] López Higuera, J. M., "Introduction to fibre optic sensing technology" capítulo 1 del libro , "Handbook of optical fibre sensing technology", John Wiley and Sons, Chichester 2002.
- [3] Crossley S. D., "Commercialisation of fibre optic sensor" capítulo 2 del libro , "Handbook of optical fibre sensing technology", John Wiley and Sons, Chichester 2002.
- [4]<https://www.arduino.cc/en/Main/ArduinoBoardUno>

Anexos

A1. DataSheet, OPF 372A

Fiber Optic Transmitter OPF372 Family



OPF372 Family

- Low Cost 850 nm LED technology
- Popular ST® style receptacle
- Pre-tested with fiber to assure performance
- Component pre-mounted and ready to use
- 35MHz operation



The OPF372 family fiber optic transmitters are high performance devices packaged for data communication links. This transmitter is an 850nm GaAlAs LED and is specifically designed to efficiently launch optical power into fibers ranging in size from 50/125µm up to 200/300µm diameter fiber. Multiple power ranges with upper and lower limits are offered which allows the designer to select a device best suited for the application.

This product's combination of features including high speed and efficient coupled power makes it an ideal transmitter for integration into all types of data communications equipment.

The mechanical design of this packaged is intended for PC Board or panel mounting. It is shipped with a lock washer, jam nut, 2#2-56 screws, and a protective dust cap.

Applications

- ◆ Industrial Ethernet equipment
- ◆ Copper-to-fiber media conversion
- ◆ Intra-system fiber optic links
- ◆ Video surveillance systems

Typical Coupled Power $I_f = 100\text{mA}, 25^\circ\text{C}$						
Fiber Size	Type	N.A.	OPF372A	OPF372B	OPF372C	OPF372D
50/125 µm	Graded Index	0.20	29µW	19µW	12.5µW	7.5µW
62.5/125 µm	Graded Index	0.28	89µW	51µW	35µW	27µW
100/140 µm	Graded Index	0.29	200µW	129µW	87µW	60µW
200/300 µm	Step Index	0.41	750µW	606µW	463µW	320µW

Absolute Maximum Ratings

$T_A = 25^\circ\text{C}$ unless otherwise noted

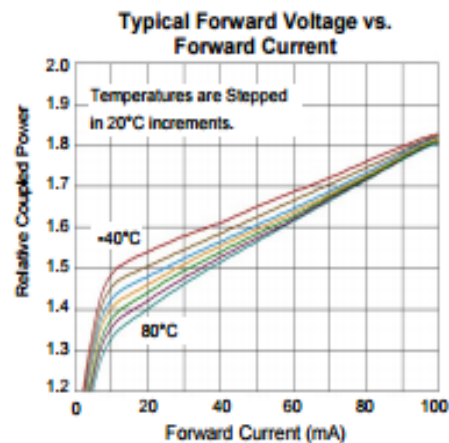
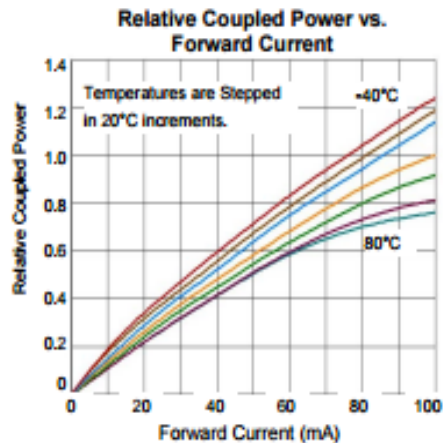
Storage Temperature Range	-55° C to +100° C
Operating Temperature Range	-40° C to +85° C
Lead Soldering Temperature ⁽¹⁾	260° C
Continuous Forward Current ⁽²⁾	100 mA
Maximum Reverse Voltage	1.0 V

Electrical/Optical Characteristics ($T_A = 25^\circ\text{C}$ unless otherwise noted)

SYMBOL	PARAMETER	MIN	TYP	MAX	UNITS	CONDITIONS
P_{OC}	Total Coupled Power 50/125 mm Fiber, NA = 0.20	OPF372A	25.0	29.0		μW $I_F = 100\text{ mA}$
		OPF372B	15.0	19.0		
		OPF372C	10.0	12.5		
		OPF372D	5.0	7.5		
V_F	Forward Voltage		1.8	2.2	V	$I_F = 100\text{ mA}$
V_R	Reverse Voltage	1.8			V	$I_R = 100\ \mu\text{A}$
λ	Wavelength	830	850	870	nm	$I_F = 50\text{ mA}$
$\Delta\lambda$	Optical Bandwidth		45	60	nm	$I_F = 50\text{ mA}$
t_r, t_f	Rise and Fall Time		6.0	10.0	ns	$I_F = 100\text{ mA}$; 10% to 90% ⁽³⁾

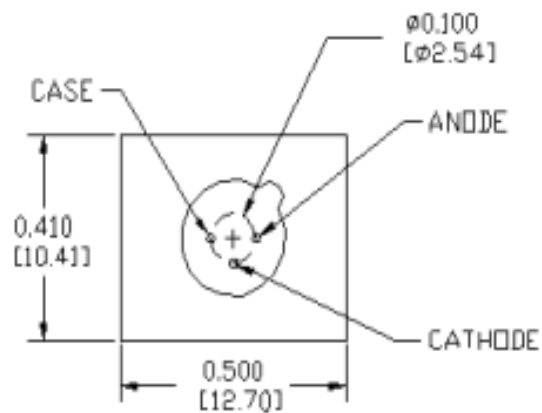
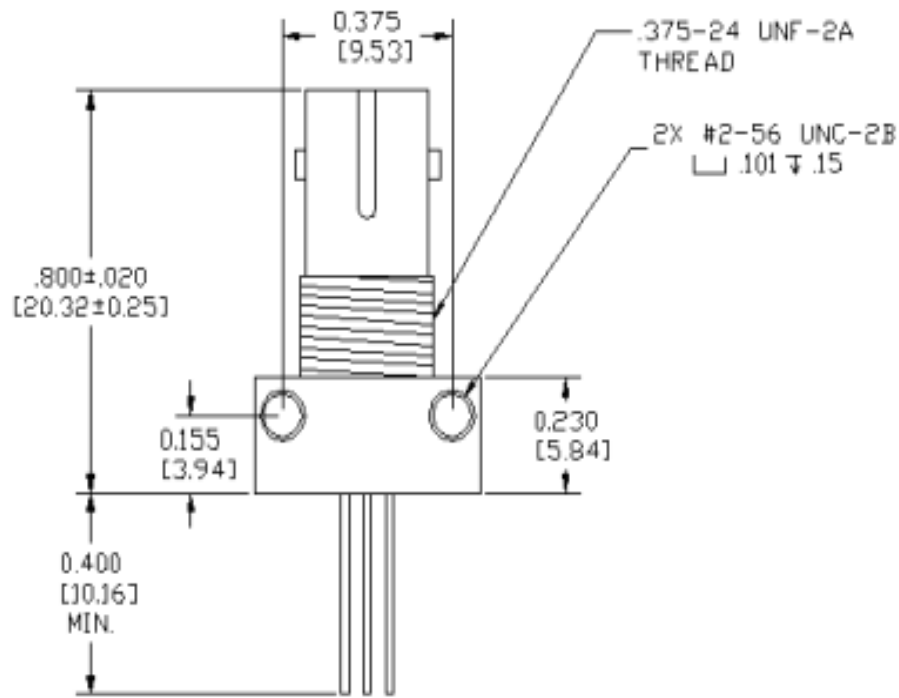
Notes:

1. Maximum of 5 seconds with soldering iron. Duration can be extended to 10 seconds when flow soldering. RMA flux is recommended.
2. De-rate linearly at 1.33mA/°C above 25°C.
3. No Pre-bias.
4. All Optek fiber optic LED products are subjected to 100% burn-in as part of its quality control process. The burn-in conditions are 96 hours at 100mA drive current and 25°C ambient temperature.



OPTEK reserves the right to make changes at any time in order to improve design and to supply the best product possible.

Mechanical Data



DIMENSIONS ARE IN INCHES (MILLIMETERS)

OPTEK reserves the right to make changes at any time in order to improve design and to supply the best product possible.

A2. DataSheet, OPF 472

Fiber Optic Detector OPF472



OPF472

- High speed, low capacitance
- Popular ST[®] style receptacle
- Pre-tested with fiber to assure performance
- Component pre-mounted and ready to use
- 35MHz operation minimum



The OPF472 is a low noise silicon PIN photodiode mounted in a low cost package for fiber optic applications. It offers fast response at moderate bias and is compatible with LED and laser diode sources in the 800-1000 nm wavelength region. Low capacitance improves signal to noise performance in typical short haul LAN applications.

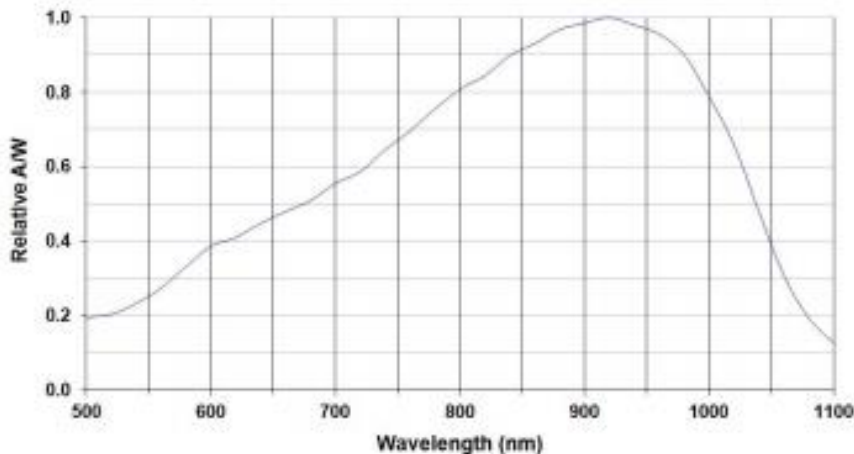
The OPF472 is designed to be compatible with multimode optical fibers from 50/125 to 200/300 microns.

Applications

- Industrial Ethernet equipment
- Copper-to-fiber media conversion
- Intra-system fiber optic links
- Video surveillance systems



Typical Responsivity



ST[®] is a registered trademark of AT&T.

OPTEK reserves the right to make changes at any time in order to improve design and to supply the best product possible.

Fiber Optic Detector

OPF472



Absolute Maximum Ratings

$T_A = 25^\circ\text{C}$ unless otherwise noted

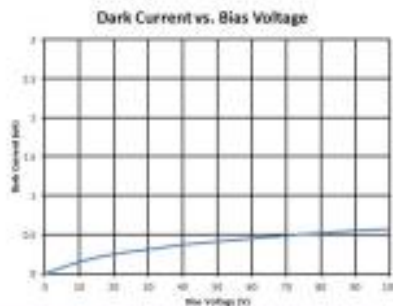
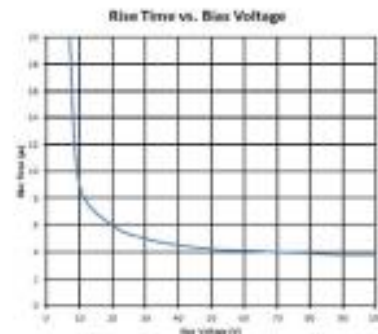
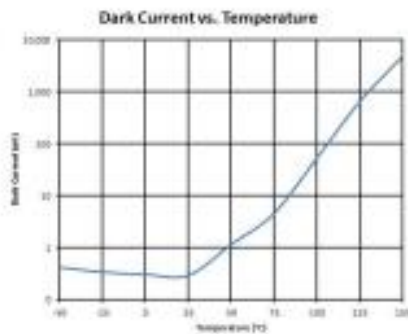
Storage Temperature Range	-55° C to +100° C
Operating Temperature Range	-40° C to +85° C
Lead Soldering Temperature ⁽¹⁾	260° C
Continuous Power Dissipation ⁽²⁾	200 mW
Maximum Reverse Voltage	100 VDC

Electrical/Optical Characteristics ($T_A = 25^\circ\text{C}$ unless otherwise noted)

SYMBOL	PARAMETER	MIN	TYP	MAX	UNITS	CONDITIONS
R	Responsivity	0.45	0.55		AW	$V_R = 5.0\text{V}$; 50/125 μm fiber; $\lambda = 850\text{nm}$
I_D	Dark Current		0.1	5.0	nA	$V_R = 5.0\text{V}$
λ_p	Peak Response Wavelength		905		nm	
t_r	Output Rise Time		6.0		ns	$V_R = 15\text{V}$; $R_L = 50\Omega$, 10%-90%
C_T	Total Capacitance		3.0		pF	$V_R = 20\text{V}$

Notes:

- Maximum of 5 seconds with soldering iron. Duration can be extended to 10 seconds when flow soldering. RMA flux is recommended.
- De-rate linearly at 2.67mW/°C above 25°C.

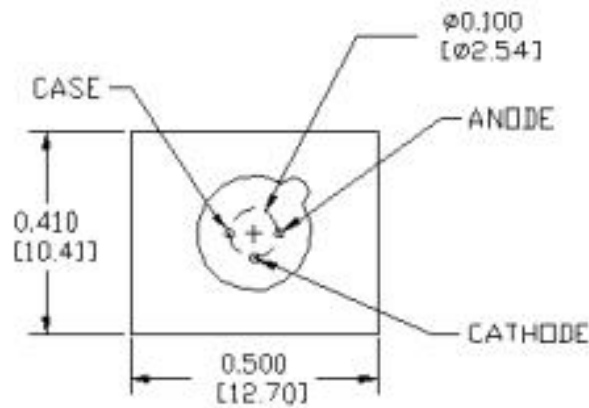
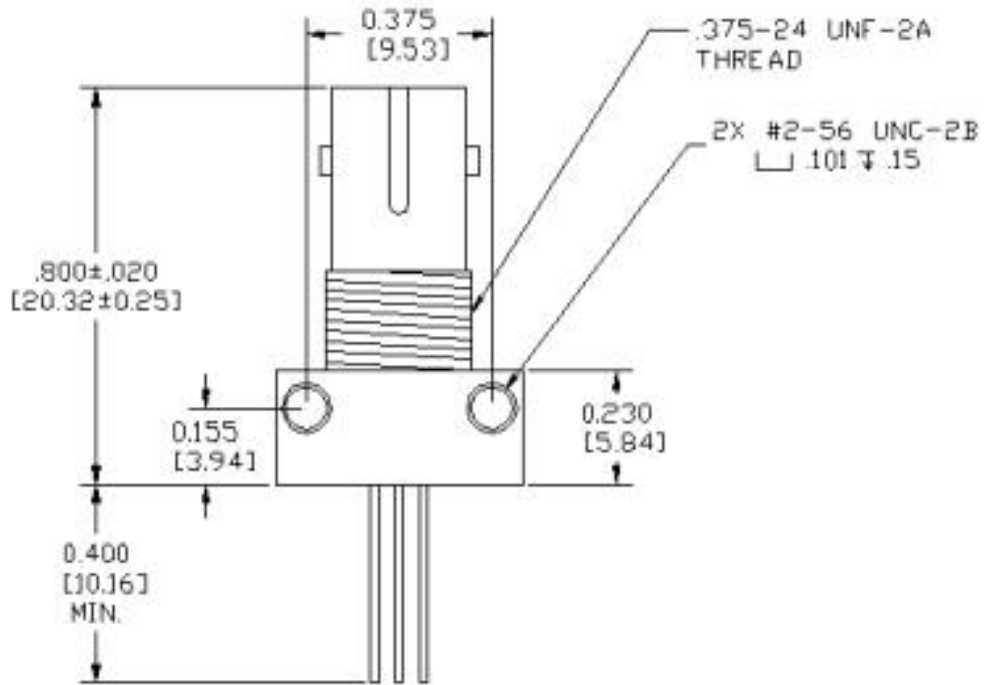


in order to improve design and to supply the best product possible.

Fiber Optic Detector OPF472



Mechanical Data




DIMENSIONS ARE IN INCHES (MILLIMETERS)

OPTEK reserves the right to make changes at any time in order to improve design and to supply the best product possible.

OPTEK Technology Inc.— 1645 Wallace Drive, Carrollton, Texas 75006
Phone: (800) 341-4747 FAX: (972) 323-2396 sensors@optekinc.com www.optekinc.com

Issue B 4/2013
Page 3 of 3



DATA SHEET

BIPOLAR ANALOG INTEGRATED CIRCUIT

μ PC358

LOW POWER DUAL OPERATIONAL AMPLIFIERS

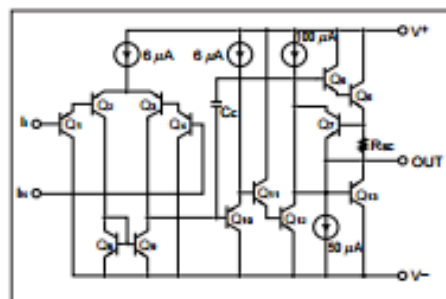
DESCRIPTION

The μ PC358 is a dual operational amplifier which is designed to operate from a single power supply over a wide range of voltages. Operation from split power supplies is also possible and the power supply current drain is very low. Further advantage, the input common-mode voltage range includes ground in the linear mode.

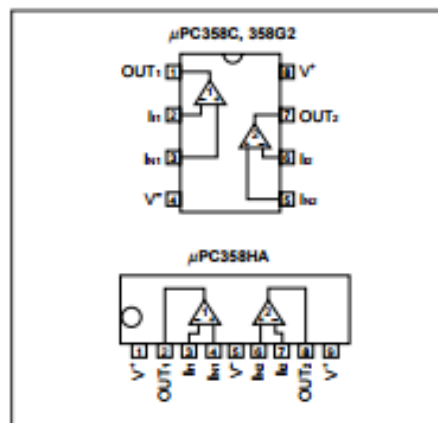
FEATURES

- Internally frequency compensation
- Wide output voltage swing V^- to $V^+ - 1.5$ V
- Common mode input voltage range includes V^-
- Wide supply voltage range
 - 3 V to 30 V (Single)
 - ± 1.5 V to ± 15 V (Split)
- Output short circuit protection

EQUIVALENT CIRCUIT (1/2 Circuit)



PIN CONFIGURATION (Marking Side)



ORDERING INFORMATION

Part Number	Package
μ PC358C	8-pin plastic DIP (300 mil)
μ PC358G2	8-pin plastic SOP (225 mil)
μ PC358HA	9-pin slim SIP

The information in this document is subject to change without notice.

ABSOLUTE MAXIMUM RATINGS (T_A = 25 °C)

Parameter		Symbol	Ratings	Unit
Voltage between V ⁺ and V ⁻ Note 1		V ⁺ - V ⁻	-0.3 to +32	V
Differential Input Voltage		V _{io}	±32	V
Input Voltage Note 2		V _i	V ⁻ -0.3 to V ⁺ +32	V
Output Voltage Note 3		V _o	V ⁻ -0.3 to V ⁺ +0.3	V
Power Dissipation	C Package Note 4	P _T	350	mW
	G2 Package Note 5		440	mW
	HA Package Note 4		350	mW
Output Short Circuit Duration Note 6			Indefinite	s
Operating Ambient Temperature		T _A	-20 to +80	°C
Storage Temperature		T _{stg}	-55 to +125	°C

- Notes**
- Reverse connection of supply voltage can cause destruction.
 - The input voltage should be allowed to input without damage or destruction independent of the magnitude of V⁺. Either input signal should not be allowed to go negative by more than 0.3 V. The normal operation will establish when the both inputs are within the Common Mode Input Voltage Range of electrical characteristics.
 - This specification is the voltage which should be allowed to supply to the output terminal from external without damage or destructive. Even during the transition period of supply voltage, power on/off etc., this specification should be kept. The output voltage of normal operation will be the Output Voltage Swing of electrical characteristics.
 - Thermal derating factor is -5.0 mW/°C when operating ambient temperature is higher than 55 °C.
 - Thermal derating factor is -4.4 mW/°C when operating ambient temperature is higher than 25 °C.
 - Pay careful attention to the total power dissipation not to exceed the absolute maximum ratings, Note 4 and Note 5.

RECOMMENDED OPERATING CONDITIONS

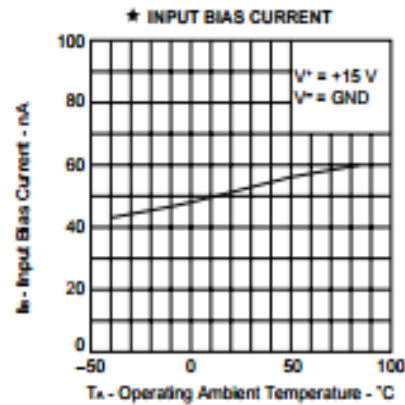
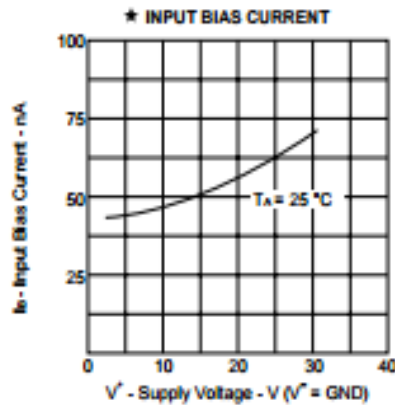
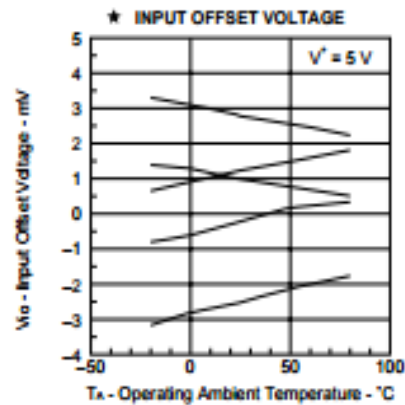
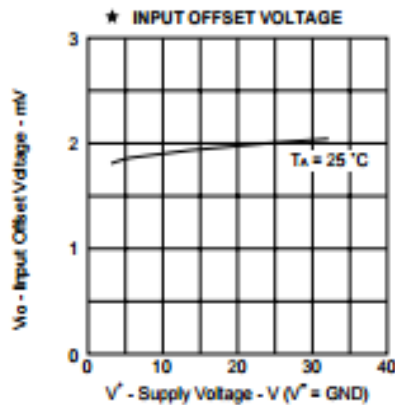
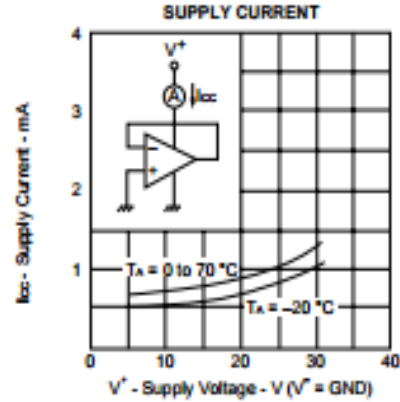
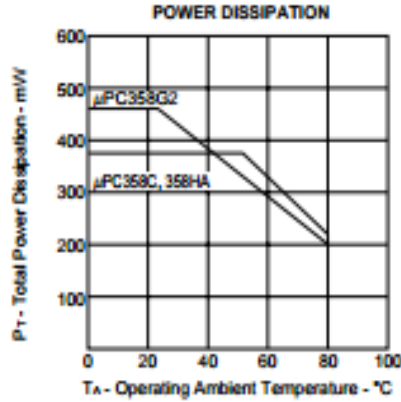
Parameter	Symbol	MIN.	TYP.	MAX.	Unit
Supply Voltage (Split)	V ⁺	±1.5		±15	V
Supply Voltage (V ⁻ = GND)	V ⁺	+3		+30	V

ELECTRICAL CHARACTERISTICS (T_A = 25 °C, V⁺ = +5 V, V⁻ = GND)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input Offset Voltage	V _{io}	R _s = 0 Ω		±2	±7	mV
Input Offset Current	I _{io}			±5	±50	nA
Input Bias Current Note 7	I _b			45	250	nA
Large Signal Voltage Gain	A _v	R _L ≥ 2 kΩ	25	100		V/mA
Supply Current	I _{cc}	R _L = ∞, I _o = 0 A, Both Amplifiers		0.7	1.2	mA
Common Mode Rejection Ratio	CMR		65	70		dB
Supply Voltage Rejection Ratio	SVR		65	100		dB
Output Voltage Swing	V _o	R _L = 2 kΩ (Connect to GND)	0		V ⁺ -1.5	V
Common Mode Input Voltage Range	V _{icm}		0		V ⁺ -1.5	V
Output Current (SOURCE)	I _{o source}	V _{in} ⁺ = +1 V, V _{in} ⁻ = 0 V	20	40		mA
Output Current (SINK)	I _{o sink}	V _{in} ⁻ = +1 V, V _{in} ⁺ = 0 V	10	20		mA
		V _{in} ⁻ = +1 V, V _{in} ⁺ = 0 V, V _o = 200 mV	12	50		μA
Channel Separation		f = 1 kHz to 20 kHz		120		dB

Note 7. Input bias currents flow out from I_C. Because each currents are base current of PNP-transistor on input stage.

TYPICAL PERFORMANCE CHARACTERISTICS ($T_A = 25^\circ\text{C}$, TYP.)



A5. Programa grabado en el microcontrolador

```
#include <OneWire.h>
#include <LiquidCrystal.h>
#include <Wire.h>

OneWire ds(A0); // on pin 10 (a 4.7K resistor is necessary)
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
int linea = 0;
int LABEL = 1;
int valor = 0;
int i;// variable de iteraciones
int s;//medida en segundos del periodo en el que se toman las medidas
int v;
int peltier = 10;
int resistencia = 9;
const int inPinA1 = A1;
int sensorValue = 0;
int Temp=12; //Parámetro de encendido de la resistencia o la celda de peltier

byte grado[8] = {
  0b00111,
  0b00101,
  0b00111,
  0b00000,
  0b00000,
  0b00000,
  0b00000,
  0b00000
};
byte termometro[8] = {
  0b00100,
  0b01010,
  0b01010,
  0b01110,
  0b01110,
  0b11111,
  0b11111,
  0b01110
};
void setup(void) {
  Serial.begin(9600);
  Serial.println("CLEARDATA");
  Serial.println("LABEL,Hora,celsius,pVolt0,linea");
```

```

/* lcd.begin(16, 2);
lcd.print("Residencia");
lcd.createChar(7, grado);
lcd.createChar(9, termometro);*/
pinMode(peltier,OUTPUT);
pinMode(resistencia,OUTPUT);
pinMode(inPinA1,INPUT);
digitalWrite(peltier,HIGH);

}
void loop(void) {
  for(i=0;i<=4999;i++){//realiza 5 iteraciones
    byte i;
    byte present = 0;
    byte type_s;
    byte data[12];
    byte addr[8];
    float celsius, fahrenheit;

    if ( !ds.search(addr)) {
      // Serial.println("No more addresses.");
      Serial.println();
      ds.reset_search();
      delay(250);
      return;
    }
    // Serial.print("ROM =");
    for( i = 0; i < 8; i++) {
      // Serial.write(' ');
      // Serial.print(addr[i], HEX);
    }
    if (OneWire::crc8(addr, 7) != addr[7]) {
      // Serial.println("CRC is not valid!");
      return;
    }
    // Serial.println();
    // the first ROM byte indicates which chip
    switch (addr[0]) {
      case 0x10:
        // Serial.println(" Chip = DS18S20"); // or old DS1820
        type_s = 1;

```

```

    break;
case 0x28:
// Serial.println(" Chip = DS18B20");
    type_s = 0;
    break;
case 0x22:
// Serial.println(" Chip = DS1822");
    type_s = 0;
    break;
default:
// Serial.println("Device is not a DS18x20 family device.");
    return;
}

ds.reset();
ds.select(addr);
ds.write(0x44); // start conversion, use ds.write(0x44,1) with parasite power on at the end
// delay(1000); // maybe 750ms is enough, maybe not
present = ds.reset();
ds.select(addr);
ds.write(0xBE); // Read Scratchpad
//Serial.print(" Data = ");
//Serial.print(present, HEX);
//Serial.print(" ");
for ( i = 0; i < 9; i++) { // we need 9 bytes
    data[i] = ds.read();
// Serial.print(data[i], HEX);
//Serial.print(" ");
}
// Serial.print(" CRC=");
// Serial.print(OneWire::crc8(data, 8), HEX);
// Serial.println();

// Convert the data to actual temperature
// because the result is a 16 bit signed integer, it should
// be stored to an "int16_t" type, which is always 16 bits
// even when compiled on a 32 bit processor.
int16_t raw = (data[1] << 8) | data[0];
if (type_s) {
    raw = raw << 3; // 9 bit resolution default
    if (data[7] == 0x10) {
        // "count remain" gives full 12 bit resolution

```

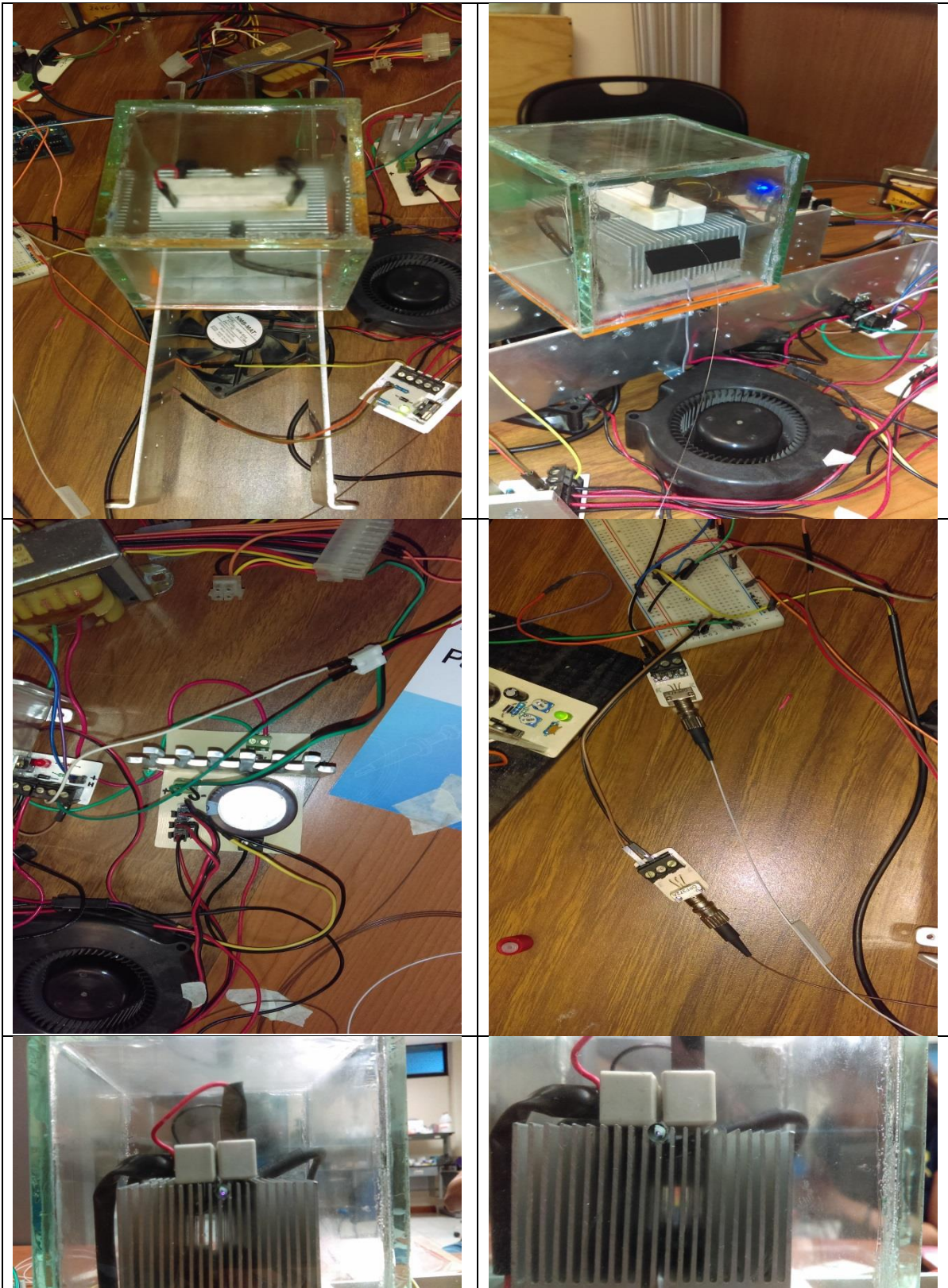
```

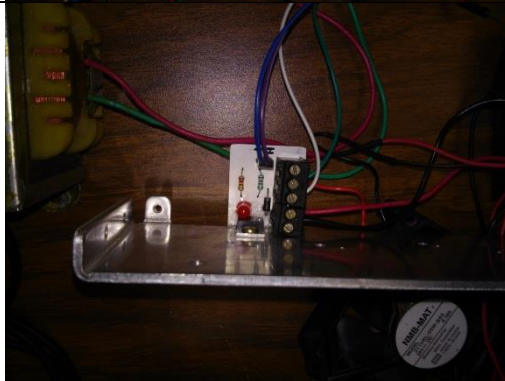
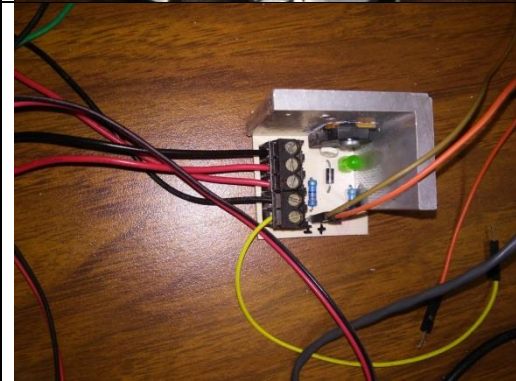
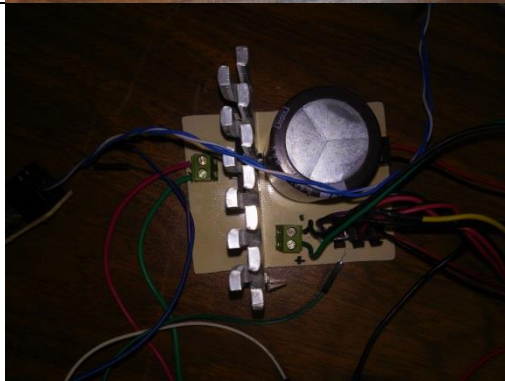
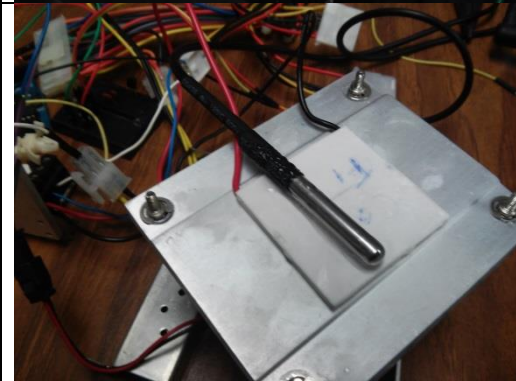
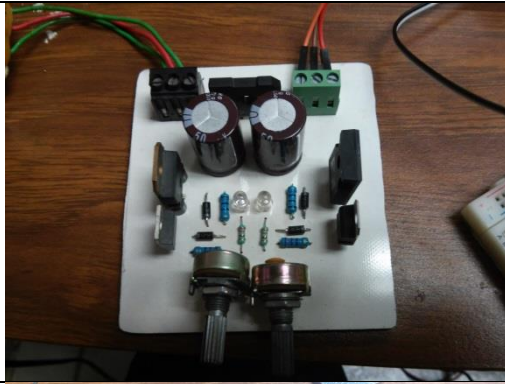
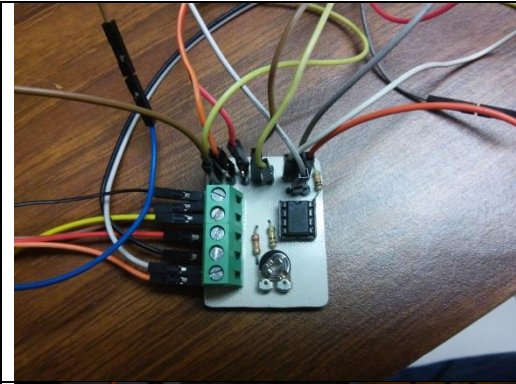
    raw = (raw & 0xFFFF) + 12 - data[6];
  }
} else {
  byte cfg = (data[4] & 0x60);
  // at lower res, the low bits are undefined, so let's zero them
  if (cfg == 0x00) raw = raw & ~7; // 9 bit resolution, 93.75 ms
  else if (cfg == 0x20) raw = raw & ~3; // 10 bit res, 187.5 ms
  else if (cfg == 0x40) raw = raw & ~1; // 11 bit res, 375 ms
  //// default is 12 bit resolution, 750 ms conversion time
}
int pinReadA1 = analogRead(inPinA1);
float pVolt0 = pinReadA1 / 1024.0 * 5.0;
celsius = (float)raw / 16.0;

if(celsius <=Temp){
  digitalWrite(peltier,LOW);
  digitalWrite(resistencia,HIGH);
}
if(celsius >= Temp+78){
  digitalWrite(peltier,HIGH);
  digitalWrite(resistencia,LOW);
}
linea++;
Serial.print("DATA,TIME,");
Serial.print(celsius);
Serial.print(",");
Serial.print(pVolt0);
Serial.print(",");
Serial.println(linea);
delay(1000);
}
}

```


A6. Imágenes Del Proyecto Terminado





Apéndice – Manual de operación.

A 1.1 Funcionamiento de la fuente de alimentación del diodo láser.

Para este proyecto se utilizaron varias fuentes de alimentación con distintos propósitos, en este apartado les hablare sobre el uso de la fuente de alimentación para el diodo láser.

Esta fuente de alimentación está formada por materiales fáciles de conseguir en el mercado y que se los presentare a continuación.

- 1.- Regulador de voltaje Lm317.
- 2.- Resistencias eléctricas (1 k Ω , 240 Ω).
- 3.- Capacitores electrolíticos (10 μ f, 2200 μ f).
- 4.- Diodos 1N4004.
- 5.- Puente de diodos KBU4M.
- 6.- Diodo led.
- 7.- Capacitor cerámico 1 μ f.
- 8.-Conectores.
- 9.- Potenciómetros (500 Ω , 5 k Ω).
- 10.- Transformador (120 VAC a 12VAC).

El funcionamiento es bastante sencillo pues lo que hay que hacer es conectar el transformador (120 VAC a 12VAC) a los conectores, el voltaje suministrado por el transformador pasara por el puente de diodos el cual rectificara el voltaje convirtiéndolo en voltaje directo (VCD), el voltaje que ahora genera el puente de diodos pasara por un filtro capacitivo para eliminar el rizo de la señal previamente rectificada y esta señal llegara a la entrada regulador de voltaje (LM317) que cuenta con unos diodos de protección para evitar corto circuitos, a la salida del regulador se conectaron unos potenciómetros que se giraran para obtener el voltaje deseado, en nuestro caso es de 1.9 V ya que es el voltaje típico que maneja nuestro diodo láser y ya para finalizar cuenta con un indicador led que nos indica si la fuente está alimentada o no en la fig. A.1 se muestra como están colocados los materiales de la fuente.

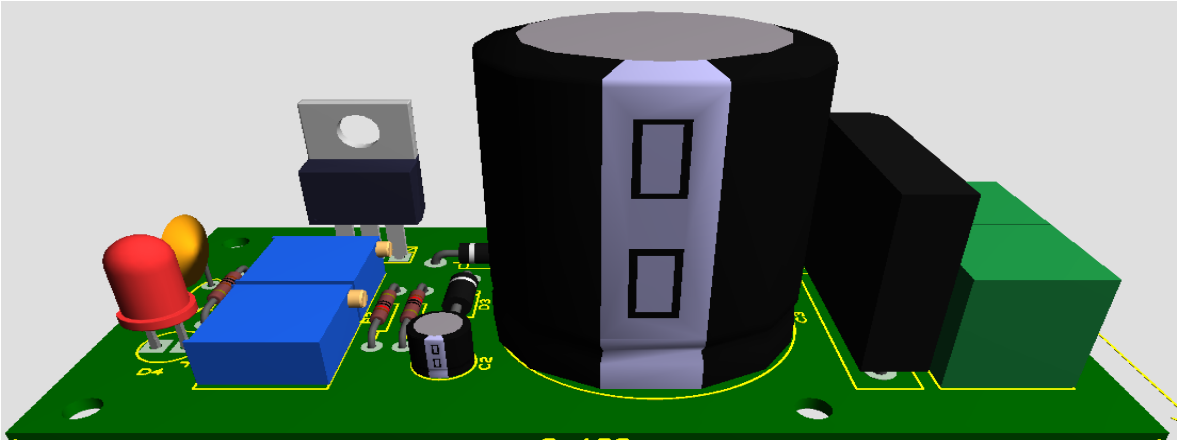


Fig. A.1 Distribución de los materiales con los que cuenta la fuente de alimentación.

A 1.2 Funcionamiento de la fuente simétrica.

En este punto se hablara sobre el funcionamiento y las partes que componen la fuente simétrica que es la que suministra el voltaje al sensor de transimpedancia.

Los materiales que se utilizaron para la fuente simétrica son los siguientes:

- 1.- Regulador de voltaje Lm317.
- 2.- Regulador de voltaje Lm337.
- 3.- Resistencias eléctricas (10 Ω , 240 Ω , 3.3 k Ω).
- 4.- Capacitores electrolíticos (1000 μ f).
- 5.- Diodos 1N4004.
- 6.- Puente de diodos KBU4M.
- 7.- Diodos led.
- 8.- Capacitores cerámicos de 100 nf.
- 9.-Conectores.
- 10.- Potenciómetros (5 k Ω).
- 11.- Transistores de potencia (TIP 35 y TIP 36)
- 12.- Transformador (120 VAC a 12VAC).

El funcionamiento es similar a la fuente de alimentación del diodo láser con la única diferencia que esta fuente maneja un lado negativo un \pm , que van desde los +15 V hasta los -15 V, y se regulan con los potenciómetros, como nuestro sensor de transimpedancia funciona con un voltaje típico de 12 V se regula hasta alcanzar los 12 V en la parte positiva y los -12 V en la parte negativa en la figura A.2 se muestra la distribución de los componentes de la fuente simétrica.

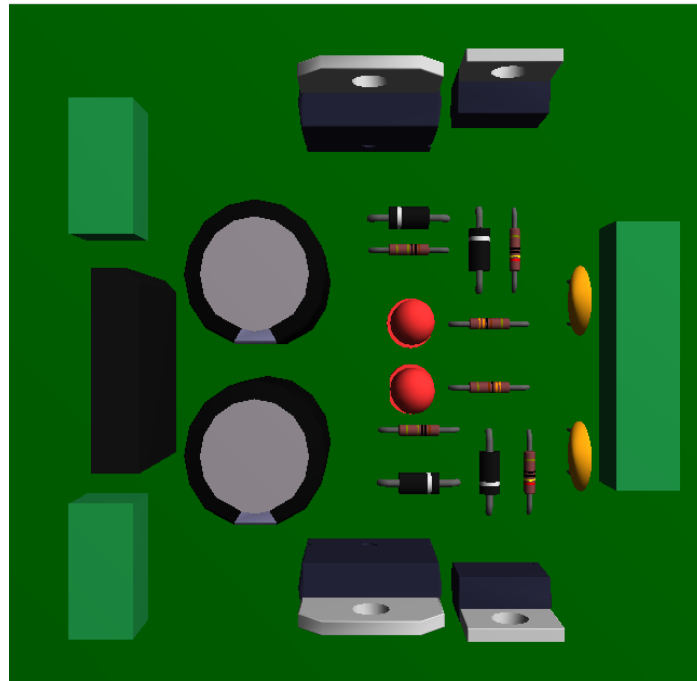


Fig. A.2 Distribución de los materiales con los que cuenta la fuente Simétrica.

A 1.3 Etapas de potencia

Las etapas de potencia son las que se encargan de suministrar más voltaje y corriente a distintos componentes.

¿Por qué diseñar una etapa de potencia?

Esta pregunta se debe a que el microcontrolador no es capaz de suministrar tanta potencia a la celda de peltier y a la resistencia eléctrica ya que estas consumen muchísimo amperaje como por ejemplo la celda de peltier consume 7 A constantes y el microcontrolador solo es capaz de suministrar 50 mA y si colocamos directamente la celda de peltier o la resistencia eléctrica a la salida de un pin del microcontrolador este se quemaría, por eso la necesidad de construir las etapas de potencia que está formado por Mosfet y en la figura A.3 se muestra como está formado.

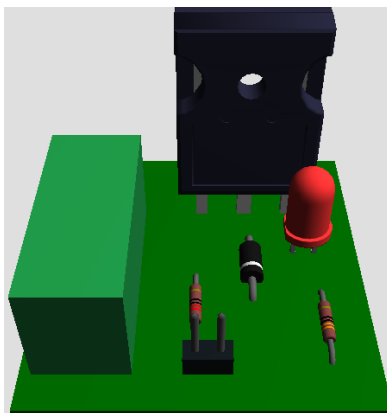


Fig. A.3 Etapa de potencia con Mosfet.

A 1.4 Sensor de transimpedancia

Para esta parte se requiere de un amplificador operacional con el arreglo de transimpedancia que nos sirva para poder convertir la corriente captada de nuestro fotodetector y convertirla en voltaje y nuestro controlador pueda captarlo e interpretarlo, el operacional a utilizar es el C358C, para poder calibrar la ganancia del operacional se utiliza la fórmula ya descrita en el proyecto una vez calculada la ganancia se ajusta el potenciómetro con el valor obtenido de la ecuación y así generar la ganancia. El circuito a utilizar se muestra en la figura A.4 que se muestra a continuación.

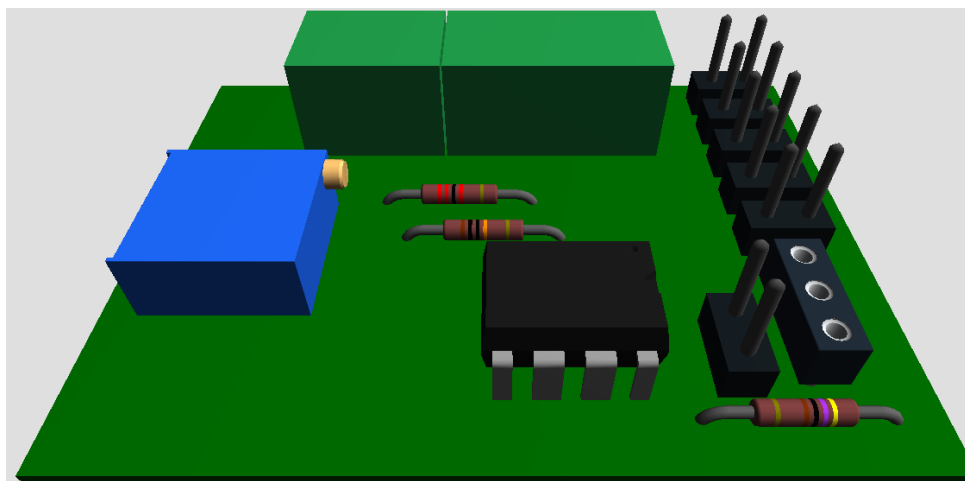


Fig. A.4 Sensor de transimpedancia con C358C.

A 1.5 interface gráfica con el usuario

Para esta parte decidí usar la herramienta MATLAB, que me sirvió para poder graficar la temperatura y el voltaje de salida del sensor de transimpedancia haciendo un vínculo con el microcontrolador en la figura A.6 se muestra la interface MATLAB.

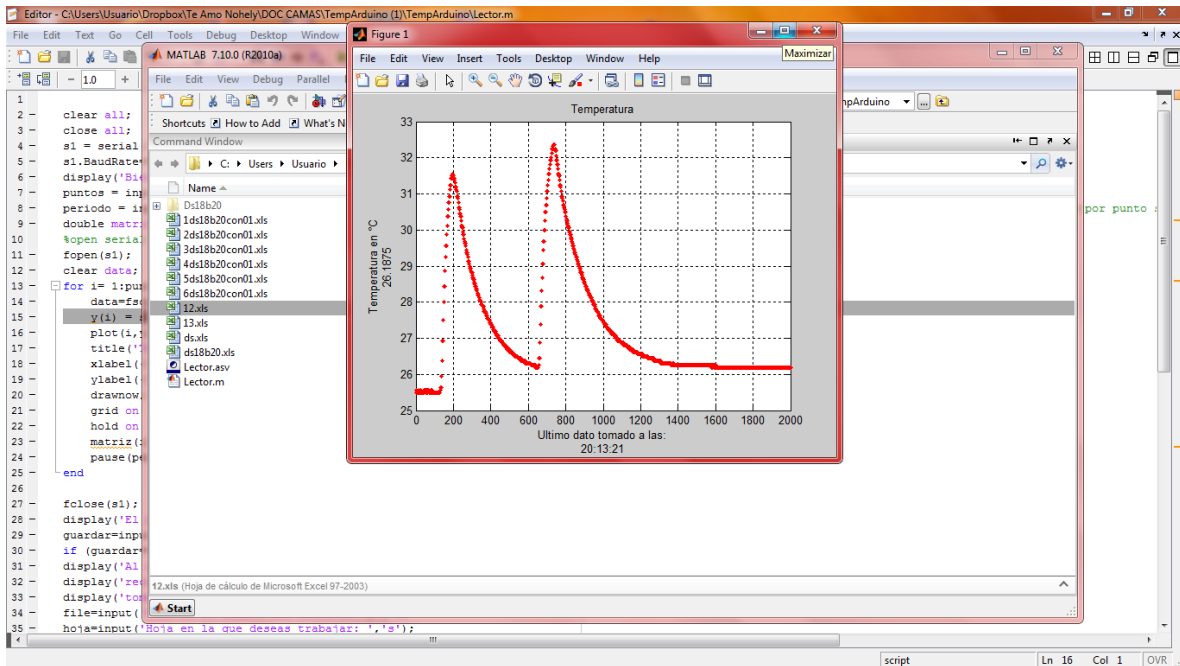


Fig. A.6 Interface gráfica MATLAB.

A 1.6 Descripción del funcionamiento del proyecto

Bueno para comenzar el monitoreo y adquisición de datos del sensor de fibra óptica lo primero que se debe hacer es seleccionar que programa cargar al microcontrolador, existen 2 programaciones Fig. A.7 y Fig. A.8, una es para caracterizar el sensor y la otra es para que la temperatura sea constante, esa temperatura la asignara el usuario por ejemplo si quiero que la temperatura sea de 40 °C se debe asignar ese valor en el programa y cargarla al microcontrolador después de hacer eso empezara el proceso automáticamente manteniéndose la temperatura asignada por el usuario cada vez que se quiera cambiar la temperatura se deberá repetir este proceso.

Como siguiente paso es abrir el programa MATLAB Fig. A.9, como siguiente paso es encender las fuentes de alimentación que son la fuente simétrica que alimenta el sensor de transimpedancia, la fuente de alimentación del diodo láser y las fuentes para el suministro de alimentación de las etapas de potencia que alimentan la resistencia eléctrica y la celda de peltier, una vez teniendo todo encendido se regresa al programa de MATLAB ahí se inicializa, al inicializar el programa le preguntara cuantos puntos desea graficar Fig. A.10 en esa parte pueden ser de 1 hasta un valor sin límite, solo que hay que tener cuidado de no poner un valor tan elevado y al final ya no quiera esperar a que termine de muestrear ya que una vez empezando el programa ya no se detendrá hasta graficar todos los puntos deseados por el usuario y la única manera de pararlo es cerrando completamente el programa pero se perderán todos los valores previamente obtenidos.

El siguiente paso es el de asignar un intervalo a la gráfica Fig. A.11 como por ejemplo usted decide graficar 1000 puntos el programa le pregunto con qué intervalos desea graficar tiene que ser distinto de 0 para que esto funcione un ejemplo puede ser 1000 puntos con intervalos de 0.1, esto quiere decir que cada muestra será de 0.1, 0.2, 0.3, Etc., hasta finalizar los 1000 puntos, una vez finalizada la muestra el programa le pedirá ingresar un “Si” siempre y cuando desee guarda los datos obtenido en una hoja de datos Excel, pero si no desea guardarlo deberá poner un “NO” y el programa terminara.

```

Projecto_Resistencia_Final_Programacion_Arduino_111
Archivo Editar Sketch Herramientas Ayuda

Projecto_Resistencia_Final_Programacion

#include <Arduino.h>
#include <LiquidCrystal.h>
#include <Wire.h>
#define LCD_RS 12 // en pin 12 (+ 4.7K resistor is necessary)
#define LCD_E 11, 5, 4, 3, 2;
int linea = 0;
int LABEL = 1;
int valor = 0;
// variable de iteraciones
// medida en segundos del periodo en el que se toman las medidas
int tv;
int peltier = 10;
int resistencia = 5;
const int INICIAL = 0;
int sensorPin = 0;
int Temp=10; //Parámetro de resistencia de la resistencia o la celda de peltier

byte grade[9] = {
  0b01111,
  0b01101,
  0b01111,
  0b01111,
  0b01000,
  0b01000,
  0b01000,
  0b01000,
  0b01000
};
  
```

Fig. A.7 Programa en la plataforma Arduino.

```

Sensor_De_Temperatura_Con_Temperatura_Constante_Arduino_113
Archivo Editar Sketch Herramientas Ayuda

Sensor_De_Temperatura_Con_Temperatura_Constante

#include <Arduino.h>
#include <LiquidCrystal.h>
#include <Wire.h>
// Resistor 10KOHM, 10KOHM, 10KOHM Temperature Display
//
// http://www.gutenberg.org/cache/epub/14441/14441.html
//
// The DallasTemperature library can do all this work for you!
// http://milesburton.com/Dallas_Temperature_Control_Library

#define LCD_RS 12 // en pin 12 (+ 4.7K resistor is necessary)
#define LCD_E 11, 5, 4, 3, 2;
int linea = 0;
int LABEL = 1;
int valor = 0;
// variable de iteraciones
// medida en segundos del periodo en el que se toman las medidas
int tv;
int peltier = 10;
int resistencia = 5;
const int INICIAL = 0;
int sensorPin = 0;
int Temp=5; //Parámetro de resistencia de la resistencia o la celda de peltier

byte grade[9] = {
  0b01111,
  0b01101,
  0b01111,
  0b01111,
  0b01000,
  0b01000,
  0b01000,
  0b01000,
  0b01000
};
  
```

Fig. A.8 Programa Arduino Temp. Constante.

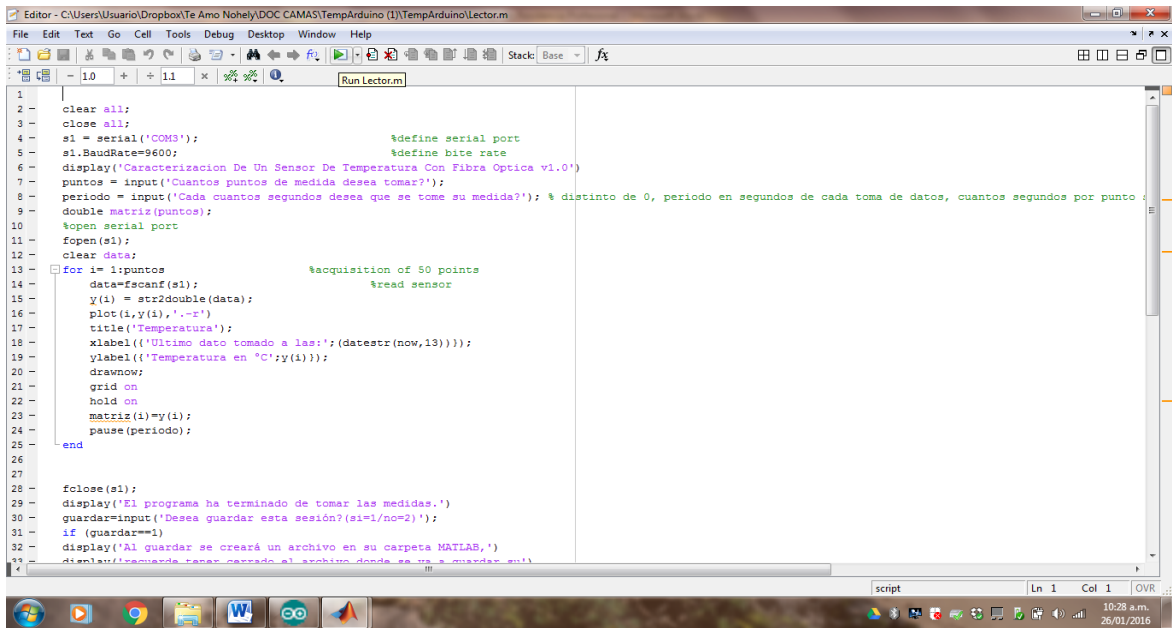


Fig. A.9 Interface usuario MATLAB

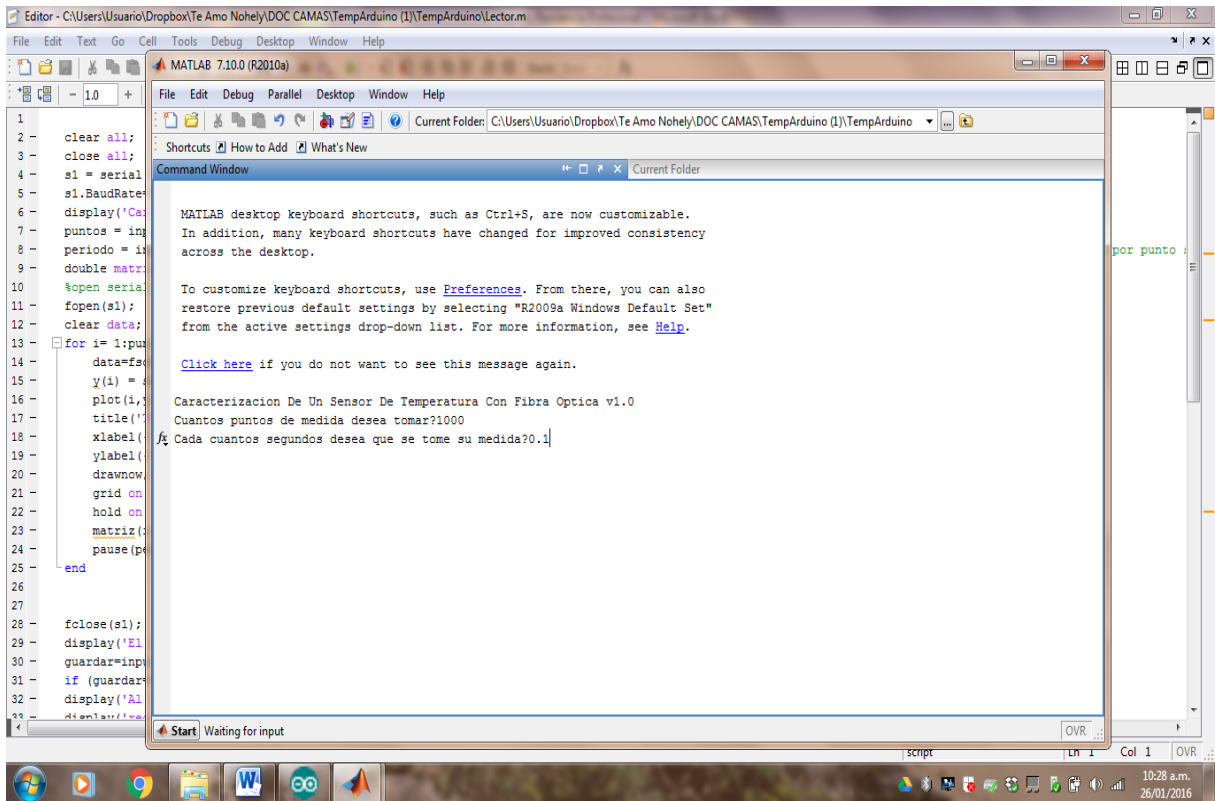


Fig. A.10 Selección de puntos a graficar

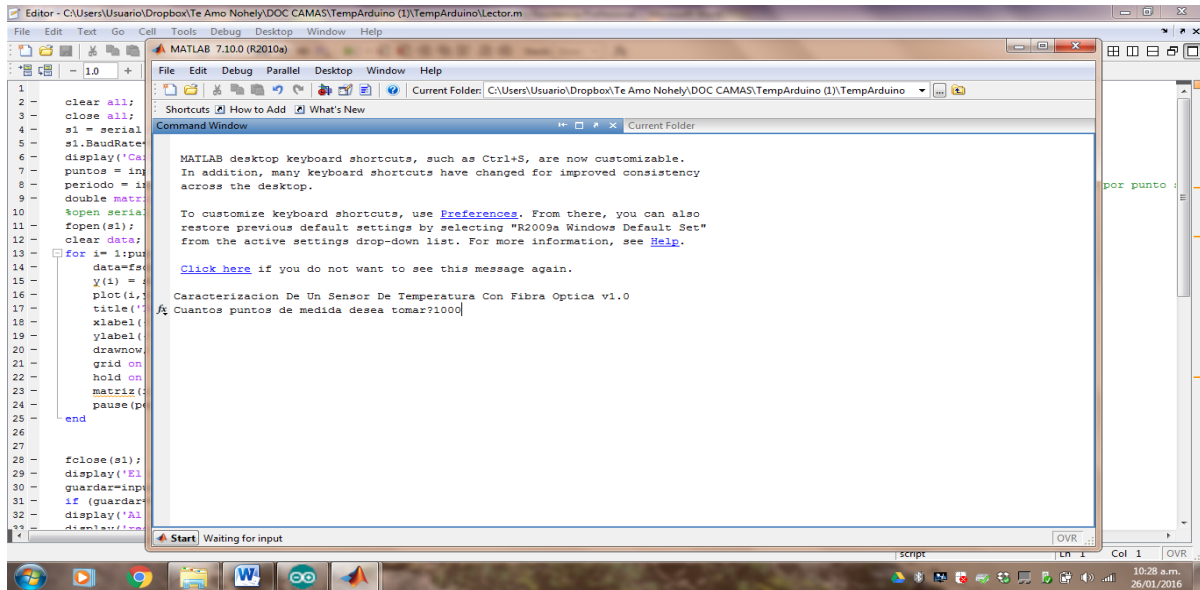


Fig. A.11 Selección de intervalo a graficar.