UNIVERSIDAD
NACIONAL
DE COLOMBIA
SEDE MEDELLÍN

Escuela de Física
Instrumentación virtual
Preinforme 1: Generar, transmitir y recibir señal AM

Docente: Ing. Rodrigo Acuña Herrera.

Presentado por:
Mauricio Ríos Hernández
Marlon Jair Garzón Piraguata

2025-1

1. objetivos

1. Adquirir emisor y fotodetector de infrarrojo (IR)
2. Generar una señal AM en LabVIEW/Python y transferirla a la salida de la DAQ/Microcontrolador
3. Convertir la señal AM de voltaje a la salida de la DAQ/Microcontrolador a una señal de corriente. El emisor de infrarrojo responde a cambios de corriente. La señal AM debe tener un nivel DC ya que el emisor no responde a valores negativos de la señal.
4. Diseñar el circuito de foto-detección. Quizás sería necesario un circuito adicional de amplificación de la señal detectada (Emisor IR). Observar la señal AM en un Osciloscopio, luego visualizarla en LabVIEW/Python utilizando la DAQ/Microcontrolador. Comparar
5. Mostrar en LabView las componentes de frecuencias (Transformada de Fourier) de la señal AM (se deben visualizar 3 frecuencias)
6. Diseñar un demodulador AM utilizando LabVIEW/Python. Esto con el objetivo de visualizar la señal de baja frecuencias (señal con el mensaje)
7. Comparar en LabView/Python la señal inicialmente generada de baja frecuencia con la señal demodulada del punto 6.

2. Teoría

2.1. Teoría de ondas electromagnéticas

Cualquier onda electromagnética puede representarse mediante su campo eléctrico o su campo magnético. Ambos campos deben cumplir con la ecuación de onda homogénea en el vacío. En este desarrollo se trabajará únicamente con el campo eléctrico por simplicidad:

$$\nabla^2 \vec{E} - \mu_0 \epsilon_0 \frac{\partial^2 \vec{E}}{\partial t^2} = 0$$

La solución general a esta ecuación es de la forma:

$$E(\vec{r}, t) = \frac{\varepsilon}{R} e^{i\left(|\vec{k}|R - \omega t + \varphi_0\right)}$$

Esta solución representa una onda esférica que se propaga desde una fuente puntual. Sin embargo, cuando se considera que el punto de observación está muy alejado de la fuente (matemáticamente en el infinito, en la práctica a varios metros), puede asumirse que el frente de onda es plano. Con esto, la expresión se transforma en:

$$E(\vec{r}, t) = \frac{\vec{\varepsilon}}{R} e^{i\left(|\vec{k}|R - \omega t + \varphi_0\right)}$$

Donde se asume que:

$$\vec{\varepsilon} = \varepsilon \hat{r}$$

Es decir, el campo eléctrico tiene una dirección fija y conocida. En este análisis, no se estudia el comportamiento vectorial del campo eléctrico, sino únicamente su magnitud, por lo que se puede expresar como:

$$E(\vec{r},t) = \varepsilon e^{i\left(\vec{k}\cdot\vec{r}-\omega t+\varphi_0\right)}$$

Si nos ubicamos en un punto fijo en el espacio, entonces el producto escalar $\vec{k}\cdot\vec{r}$ se vuelve constante, y puede absorberse en la fase inicial:

$$E(t) = \varepsilon e^{i(-\omega t+\varphi_0)}$$

Aplicando la identidad de Euler:

$$e^{ix} = \cos x + i\sin x$$

Se toma la parte real para obtener una señal físicamente medible:

$$E(t) = \varepsilon\cos(-\omega t+\varphi_0)$$

Si se considera que la fase inicial $\varphi_0 = 0$, y utilizando que $\cos(-\omega t) = \cos(\omega t)$, se obtiene:

$$E(t) = \varepsilon\cos(\omega t)$$

El voltaje entre dos puntos A y B, en presencia de un campo eléctrico uniforme, se define como:

$$V(t) = V_B - V_A = -\int_A^B \vec{E}\cdot d\vec{l}$$

Si el campo es constante y uniforme a lo largo de una distancia l, entonces:

$$V(t) = lE(t)$$

Sustituyendo la expresión del campo eléctrico:

$$V(t) = l\varepsilon\cos(\omega t)$$

Se puede definir $V_{\text{máx}} = l\varepsilon$, con lo que finalmente se obtiene:

$$V(t) = V_{\text{máx}}\cos(\omega t)$$

Y recordando que $\omega = 2\pi f$, la forma más utilizada en teoría de señales es:

$$V(t) = V_{\text{máx}}\cos(2\pi f t)$$

Esta expresión representa una señal portadora senoidal de frecuencia f, que es la base para aplicaciones como la modulación en amplitud (AM), en donde se varía la amplitud de esta señal de acuerdo con una señal de información. Es importante resaltar que la última expresión se puede escribir también con seno sin cambiar su sentido físico (el coseno es un seno desplazado y visceversa)

2.2. Teoría de la modulación y Demodulación AM

En comunicaciones, una señal de información de baja frecuencia no puede transmitirse eficientemente a largas distancias por sí sola. Por esta razón, se utiliza una técnica llamada **modulación**, que consiste en modificar una señal de alta frecuencia (llamada portadora) en función de la señal de información. Una de las técnicas más simples y ampliamente utilizadas es la **modulación en amplitud** (AM).

Se definen las siguientes señales:

- $V_m(t)$: señal de mensaje o envolvente (frecuencia baja).

- $V_c(t)$: señal portadora (frecuencia alta).

$$V_m(t) = \cos(\omega_m t)$$
$$V_c(t) = \cos(\omega_c t)$$

Donde $\omega_m = 2\pi f_m$ es la frecuencia angular de la señal de mensaje y $\omega_c = 2\pi f_c$ la de la portadora. Se asume que $f_c \gg f_m$, es decir, que la frecuencia de la portadora es mucho mayor que la de la señal de mensaje.

La señal modulada en amplitud se define como:

$$V_{AM}(t) = V_c\,[1 + m\cos(\omega_m t)]\cos(\omega_c t)$$

donde el parámetro m es el **índice de modulación**, definido como:

$$m = \frac{V_m}{V_c}$$

Esta fórmula muestra que la amplitud de la portadora se modifica según la señal de información. Para comprender su contenido espectral, se aplica una expansión trigonométrica:

$$V_{AM}(t) = V_c\cos(\omega_c t) + \frac{mV_c}{2}\left[\cos((\omega_c - \omega_m)t) + \cos((\omega_c + \omega_m)t)\right]$$

Esta expresión revela que la señal modulada contiene tres componentes de frecuencia:

- Una componente en f_c (frecuencia portadora)
- Una componente en $f_c - f_m$ (banda lateral inferior)
- Una componente en $f_c + f_m$ (banda lateral superior)

Desde el punto de vista del análisis de Fourier, el espectro de esta señal muestra tres picos:

$$\text{Espectro: } \delta(f - f_c) + \delta(f - (f_c - f_m)) + \delta(f - (f_c + f_m))$$

Este desplazamiento espectral permite transmitir información de baja frecuencia a través de frecuencias más altas que son más aptas para propagarse en el espacio libre o por medios eléctricos.

Para recuperar la señal original $V_m(t)$, se utiliza un proceso llamado **demodulación**. En modulación AM, una técnica común es el *detector de envolvente*, que extrae la envolvente de la señal modulada. Esta envolvente corresponde a:

$$\text{Envolvente: } V_c\,[1 + m\cos(\omega_m t)]$$

Aplicando un filtro paso bajo que elimine la componente de alta frecuencia ω_c, se obtiene nuevamente la señal de mensaje:

$$V_m(t) = \cos(\omega_m t)$$

La siguiente imagen muestra el circuito más básico para demodular una señal modulada en amplitud (AM): un **detector de envolvente**, también conocido como *detector de pico*. Este circuito permite extraer la información contenida en la señal AM de manera sencilla y eficiente.

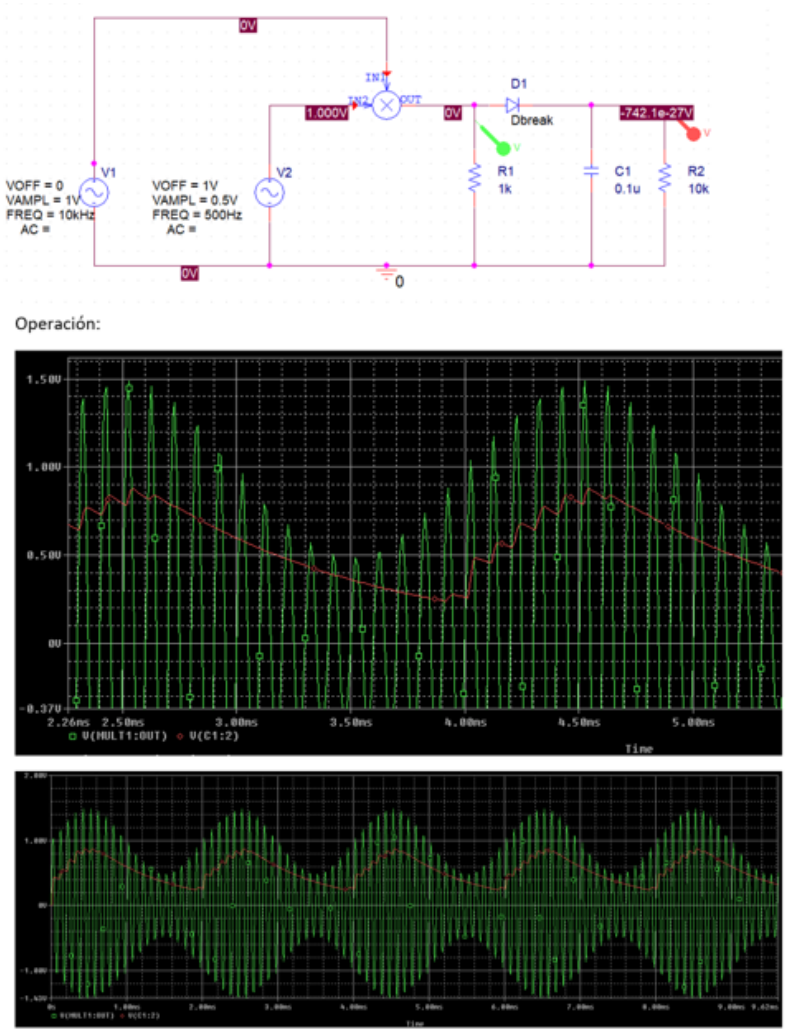


Figura 1: Detector de envolvente.

El circuito está compuesto por tres elementos fundamentales:

- **Diodo (D):** actúa como un *rectificador*, permitiendo el paso únicamente de una parte del ciclo de la señal AM (por lo general, los semiciclos positivos).
- **Condensador (C):** se carga rápidamente al valor pico de cada ciclo rectificado. Su función es suavizar la señal, siguiendo el contorno de la envolvente.
- **Resistencia (R):** permite la descarga progresiva del condensador entre picos sucesivos. Esto evita que el condensador permanezca cargado de forma indefinida. El producto RC (constante de tiempo) debe seleccionarse cuidadosamente para seguir la variación lenta de la envolvente, sin oscilar con la frecuencia de la portadora.

En la parte inferior de la imagen se muestran las formas de onda:

- La señal de entrada V_{pAM}: una señal AM compuesta por una portadora de alta frecuencia cuya envolvente representa la información útil.
- La señal de salida V_{dp}: corresponde a la envolvente de la señal de entrada, es decir, la señal de baja frecuencia que se deseaba transmitir originalmente.

Funcionamiento general:

1. La señal AM ingresa al diodo, que bloquea los semiciclos negativos.
2. El condensador se carga con los valores máximos (picos) permitidos por el diodo.
3. La resistencia permite que el condensador se descargue entre un pico y otro, generando así una señal continua que sigue la envolvente.

Este tipo de demodulación es adecuado para muchas aplicaciones prácticas como la radio AM comercial. Sin embargo, su efectividad depende del valor apropiado de RC, de una buena relación señal/ruido y de que no haya sobre-modulación (es decir, $m \leq 1$).

2.3. Led infrarrojo

Es un diodo emisor de luz que emite radiación electromagnética en la región del infrarrojo cercano del espectro (invisible al ojo humano). Es una fuente común en sistemas de comunicación óptica de corto alcance. Cuando una corriente directa (DC) pasa por un LED, los electrones cruzan la unión P-N del semiconductor y emiten fotones. En un LED infrarrojo, esos fotones tienen longitudes de onda mayores que la luz visible, típicamente en 850 nm o 940 nm.

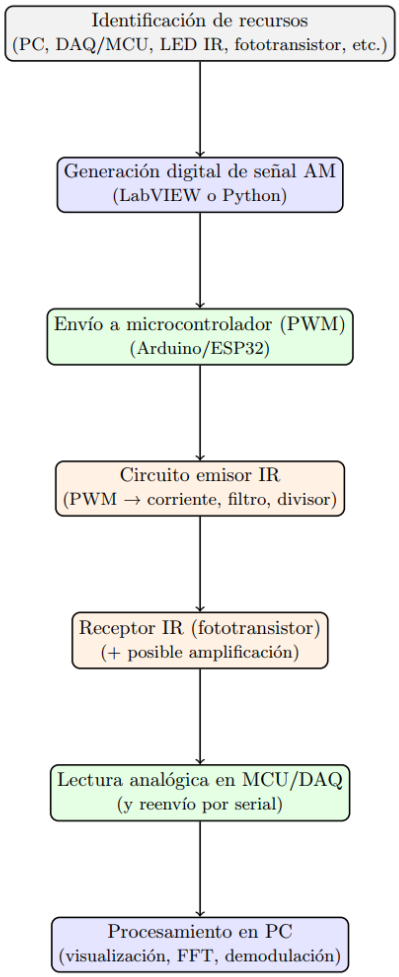
2.4. Fotodiodo

Es un dispositivo semiconductor que opera en inversa (polarización inversa), donde la corriente fluye solo cuando recibe luz infrarroja. No emite luz; en cambio, detecta fotones y los convierte en corriente.

2.5. Fototransistor

funciona como un transistor bipolar donde la corriente base es generada por la luz infrarroja. Al incidir la luz, se activa el transistor y amplifica la señal.

3. Diagrama de bloques con explicaciones de la solución al problema



Para cumplir los objetivos propuestos en la práctica de modulación AM con emisión y detección por infrarrojo, se plantea una solución dividida en etapas consecutivas:

1. Identificación de elementos físicos necesarios:

Se requieren los siguientes componentes:

- Computador (para generación y procesamiento de señales).
- Tarjeta DAQ o microcontrolador (Arduino/ESP32).
- Emisor IR (diodo LED IR).
- Fotodetector (fototransistor).
- Resistencias, capacitores, potenciómetros.
- Cables, protoboard u otros medios de conexión.

2. Generación digital de señal AM:

Existen dos formas de generar la señal AM:

- **LabVIEW:** mediante bloques de simulación de señales, se puede construir una portadora y una señal de mensaje (envolvente), y multiplicarlas para obtener la señal modulada. Esta señal debe ser convertida de tipo *dynamic* a *integer*, luego a *string*, para enviarla por comunicación serial mediante la librería VISA.
- **Python:** se utiliza la ecuación matemática de modulación AM:

$$V_{AM}(t) = V_c\,[1 + m\cos(\omega_m t)]\cos(\omega_c t)$$

  Se genera un arreglo de valores digitalizados (muestreados) y se transmiten por el puerto serial usando la librería `pyserial`.

3. Transmisión a la salida del microcontrolador:

En el caso de trabajar con microcontroladores sin salidas analógicas (como Arduino UNO), la señal se transmite vía PWM (modulación por ancho de pulso). Esto implica:

- Normalizar los datos entre 0 y 255.
- Asociar cada valor a un ciclo de PWM (duty cycle proporcional).
- Programar el microcontrolador para recibir datos por serial y aplicar la salida PWM correspondiente.

4. Circuito emisor IR:

El fotodiodo IR no responde a voltajes negativos, por lo que la señal debe tener un nivel DC. Se construye el siguiente circuito:

- Se utiliza un divisor de voltaje para reducir el voltaje de la salida PWM (típicamente 5V) a un rango aceptable para el diodo IR.
- Se coloca un condensador en paralelo al diodo para filtrar el ruido generado por la conmutación del PWM.
- El diodo se polariza directamente, respondiendo a los cambios de corriente de la señal AM.

5. Circuito receptor:

El fototransistor detecta la luz infrarroja modulada, generando una corriente proporcional. La señal resultante puede tener amplitud reducida:

- Se puede amplificar usando un amplificador operacional en configuración no inversora o seguidor de voltaje.
- La señal se dirige a una entrada analógica del DAQ o del microcontrolador.

6. Lectura y transmisión al PC:

El microcontrolador o DAQ recibe la señal modulada, la digitaliza (ADC), y la envía al computador por comunicación serial para ser procesada.

7. Procesamiento y visualización:

Con la señal digitalizada en el PC, se puede:

- Visualizar la señal cruda en el dominio del tiempo.
- Realizar su Transformada de Fourier para identificar las tres componentes de frecuencia (portadora y bandas laterales).
- Implementar un demodulador AM (por ejemplo, rectificación + filtrado o demodulación matemática).
- Comparar la señal de mensaje original con la demodulada.

4. Circuitos y simulaciones necesarias

4.1. Generar señal AM en LabVIEW/python y transferirla a la salida del DAQ/Microcontrolador.

Para generar la señal AM , se plantea dos alternativas: Generar la señal con Labview, usando la librerías VISA, para comunicación serial. Lo que se pretende es generar la señal en labview, realizar un muestreo a esta señal y enviar el valor como string, para que el microcontrolador/DAQ logre leer este valor en el buffer.

Para generar la señal se simula dos señales sinusoidales, (portadora y envolvente), en este caso se genera la portadora de 100 Hz (se trabaja con señales de baja frecuencia para visualizar el comportamiento con un LED, conectado en un microcontrolador, en este caso un Arduino Uno, lo que posteriormente se cambiará con un fotodiodo) y una envolvente de 10 HZ. Se muestrea con una frecuencia de 10000 Hz, también buscando que sea una frecuencia menor a la que el microcontrolador pueda tomar los valores.

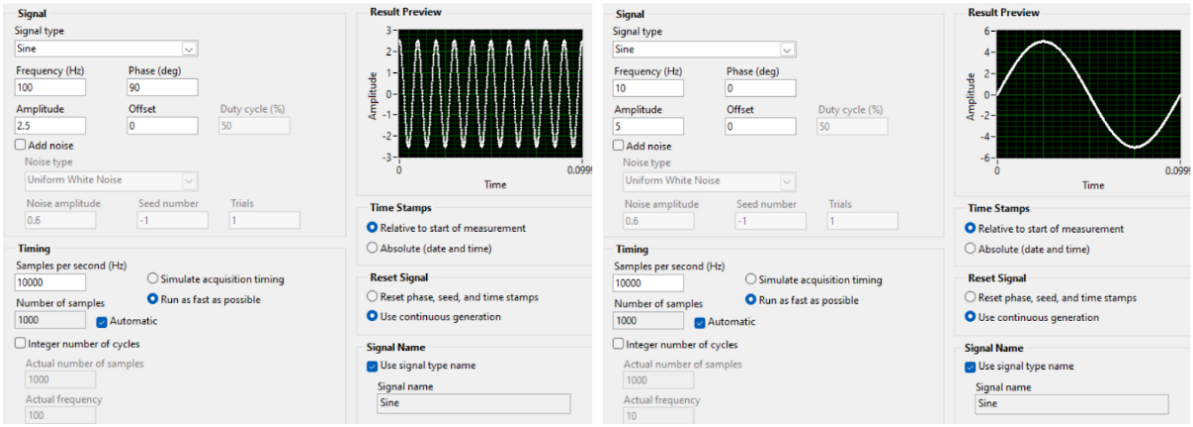


Figura 2: Señales envolvente y portadora Labview.

Obteniendo así las siguientes señales:

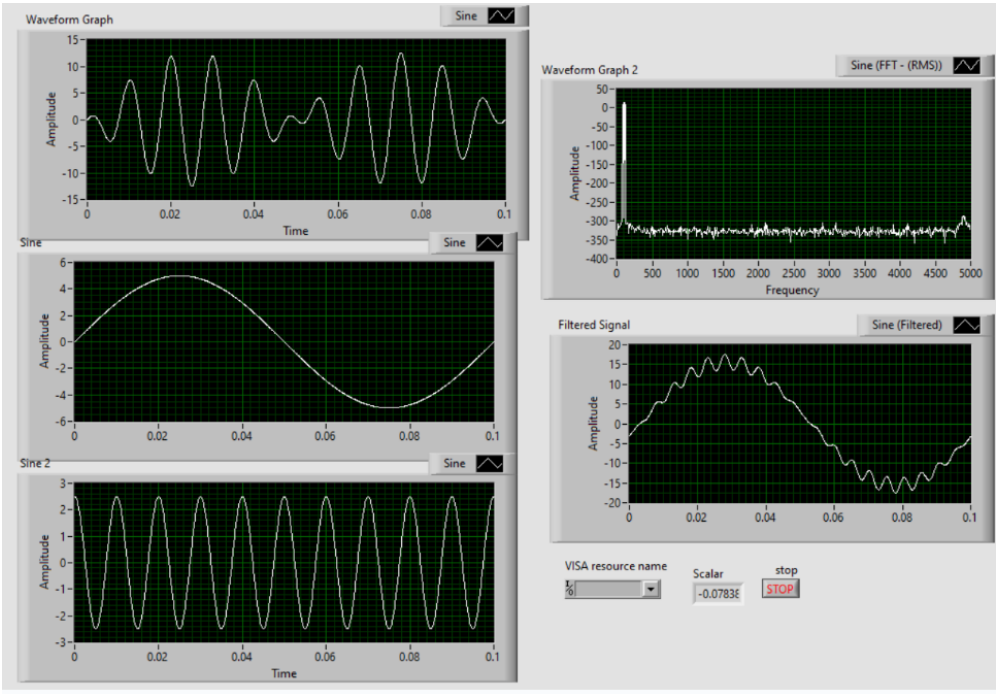


Figura 3: Señales obteniedas y generadas en Labview.

El valor que sale del multiplicador es de tipo Dynamic value, por lo que se debe transformar en tipo numérico, para después se convierte este dato numérico en un string para enviarlo por comunicación serial, al buffer del microcontrolador. El diagrama de bloques del Labview para realizar lo anterior se presenta en la siguiente imagen

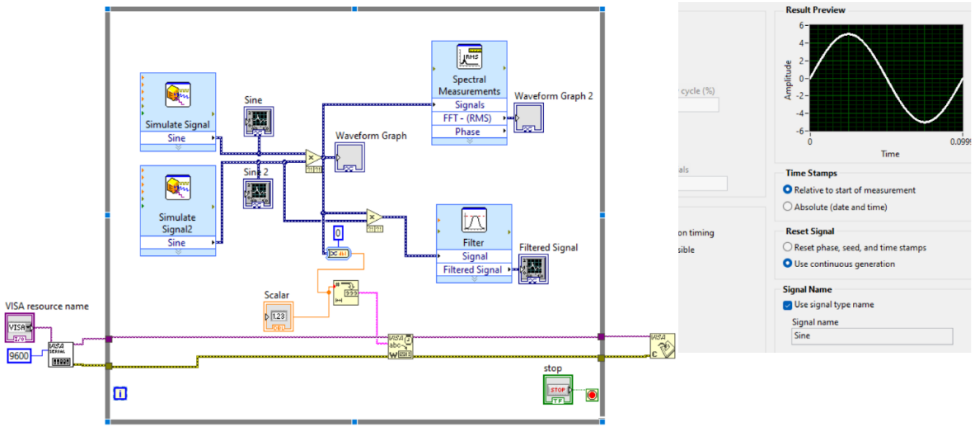


Figura 4: Diagrama de bloques de señales en Labview.

Para identificar la señal que está llegando al arduino cómo el puerto para comunicación serial está ocupado por la comunicación con Labview, el cuál está enviando los valores de la señal simulada, no es posible ver estos valores en el monitor serial del IDLE de Arduino, por lo que una opción para visualizar la señal es generar una señal PWM la cual varía el duty dependiendo del valor ingresado en la comunicación serial, claramente normalizando y tomando el valor absoluto de este.

Para lograr esto, ahora se debe programar el arduino, en este caso se utiliza el siguiente script:

```
void setup() {
  Serial.begin(9600);   // Asegúrate de que coincida con LabVIEW
  pinMode(11, OUTPUT);  // Pin 11 como salida PWM
}

void loop() {
  if (Serial.available()) {
    String input = Serial.readStringUntil('\n');
    input.trim();  // Elimina espacios o saltos de línea

    float valor = input.toFloat();
    float valorAbs = abs(valor);  // Toma el valor absoluto

    // Normaliza: 0 a 13 → 0 a 255
    int pwmValue = mapFloat(valorAbs, 0.0, 13.0, 0.0, 255.0);
    pwmValue = constrain(pwmValue, 0, 255);  // Asegura que no exceda el rango

    analogWrite(11, pwmValue);  // Aplica señal PWM
  }
}

// Función para hacer mapeo con valores decimales
int mapFloat(float x, float in_min, float in_max, float out_min, float out_max) {
  return (int)((x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min);
}
```

Figura 5: Diagrama de bloques de señales en Labview.

Ahora, analizando otra alternativa, esto generando la señal con python, se debe usar la librería serial para acceder al puerto serial. En este caso se apela directamente a la expresión de la señal AM modulada, donde simplemente se deben conocer las amplitudes y frecuencias de las señales de la envolvente y de la portadora, para obtener la expresión matemática, de esta forma se genera la señal a enviar. En la siguiente figura, se enseña el programa usado:

```python
import numpy as np
import matplotlib.pyplot as plt
import serial
import time

# Parámetros de la señal
fs = 1000       # Frecuencia de muestreo (Hz)
duracion = 2    # Duración en segundos
fc = 60         # Frecuencia de la portadora (Hz)
fm = 10         # Frecuencia de la envolvente (Hz)
Am = 0.5        # Amplitud de la envolvente
Ac = 1          # Amplitud de la portadora
offset = 2   # Offset para mantener la señal en positivo

# Vector de tiempo
t = np.linspace(0, duracion, int(fs * duracion))

# Señales
envolvente = Am * np.sin(2 * np.pi * fm * t)
portadora = Ac * np.sin(2 * np.pi * fc * t)
m = Am / Ac
senal_am = Ac * (1 + m * np.sin(2 * np.pi * fm * t)) * np.sin(2 * np.pi * fc * t)

# Señal con offset
senal_am_offset = senal_am + offset

# Normalización para PWM
senal_am_normalizada = np.interp(senal_am_offset, (min(senal_am_offset), max(senal_am_offset)), (0, 255))

# Envío por puerto serial (ajusta COMX)
puerto = serial.Serial('COM5', 9600)
time.sleep(2)

for valor in senal_am_normalizada:
    mensaje = f"{int(valor)}\n"
    puerto.write(mensaje.encode())
    time.sleep(1/fs)

puerto.close()
print("Señal enviada completamente.")
```

Figura 6: script para envío de señal en python.

Usando Python también se grafica las señales generadas y las que se enviarán al Arduino, en este caso se observa claramente la señal que se está transmitiendo al arduino y el comportamiento que tendría en el pin 11, la cuál representa bien la señal modulada (si la frecuencia del muestreo por parte del arduino es alta). Es importante resaltar que la frecuencia de muestreo desde el computador (Python), en este caso es de 1000 Hz, esta frecuencia debe ser suficiente para que el arduino, lea el dato y varíe el duty de las señal PWM (al fin de cuentas se está generando una señal cuadrada pero se le está variando la frecuencia para obtener un comportamiento similar a la de la envolvente en la salida)y se manifieste por un tiempo considerable para que la señal de salida sea aproximadamente la señal AM. La señal que envía el anterior script se presenta a continuación:

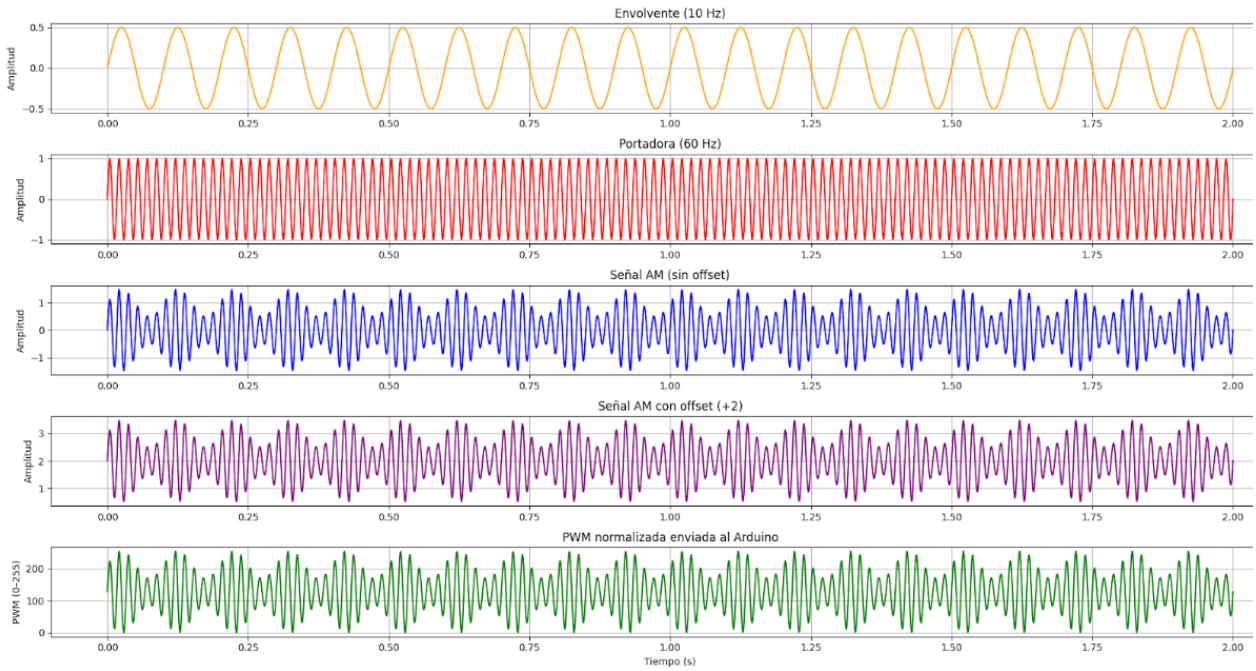


Figura 7: Señal enviada por puerto serial y generada usando Python.

Sacando la transformada de Fourier se observan 3 frecuencias características de una señal AM, esto es, las bandas laterales y la frecuencia de la portadora:

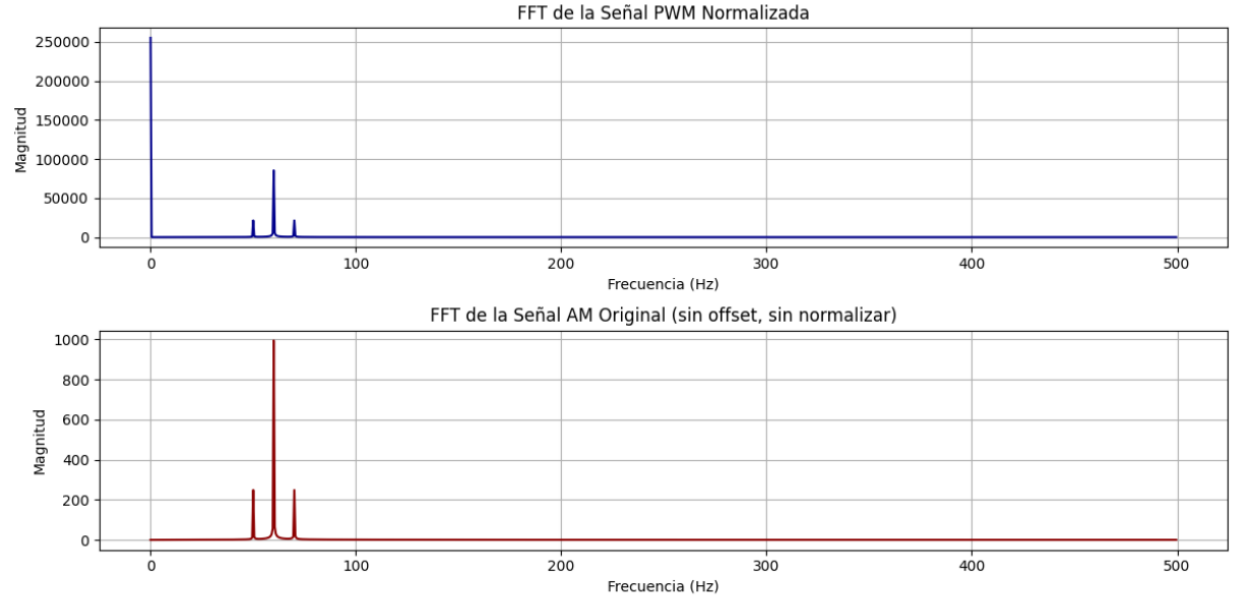


Figura 8: Espectro de señal enviada por puerto serial y generada usando Python.

En la gráfica superior, el pico en frecuencia 0 hace referencia al offset impuesto; a pesar de esto igual conserva los 3 picos característicos de la señal AM

Ahora, para que el arduino lea por el puerto serial, se debe programar con el siguiente código:

```ino
void setup() {
  Serial.begin(9600);
  pinMode(11, OUTPUT);
}

void loop() {
  if (Serial.available()) {
    String input = Serial.readStringUntil('\n');
    input.trim();
    int pwm = input.toInt();

    // Asegura que esté entre 0 y 255
    pwm = constrain(pwm, 0, 255);

    analogWrite(11, pwm);  // Aplica PWM al pin 11
  }
}
```

Figura 9: Código para lectura de puerto serial en arduino.

Se resalta de la figura anterior que cómo los valores ya estaban normalizados desde python, el arduino no realiza ninguna conversión directamente, solo toma el valor que está en un string y lo transforma a entero para así cambiar el duty de la señal que se está enviando por el pin 11, si conectamos un led a este, se verá que la luz emitida por el diodo led aumenta gradualmente y luego disminuye de esta misma manera (forma sinusoidal).

4.2. Circuito de emisión y recepción en infrarrojo

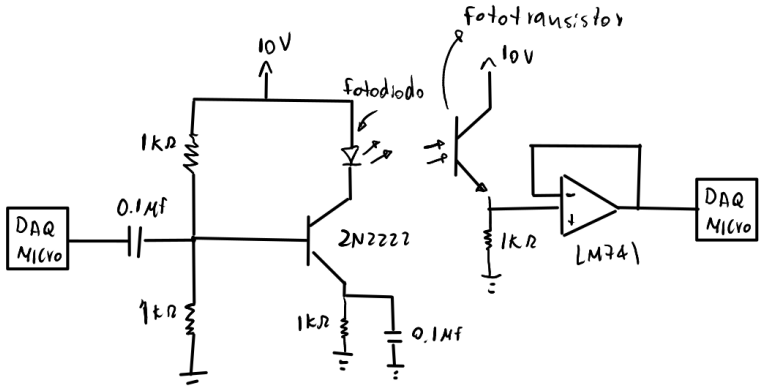


Figura 10: Circuito de emisión y recepción.

El sistema de transmisión y recepción mostrado en la figura puede entenderse como un circuito de optoacoplamiento que permite transmitir señales analógicas mediante modulación de luz infrarroja, utilizando un fotodiodo y un fototransistor como medio de acople óptico entre la etapa emisora y receptora. Este tipo de circuito es especialmente útil para evitar acoplamientos directos eléctricos entre diferentes partes del sistema, lo que mejora el aislamiento y reduce el ruido eléctrico.

En la parte izquierda de la figura se encuentra el circuito emisor, cuyo propósito principal es convertir la señal de voltaje proveniente del DAQ o microcontrolador en una señal de corriente que pueda excitar al fotodiodo. Esto se logra mediante el uso de un transistor tipo NPN (2N2222), configurado como un amplificador de corriente. La señal modulada, acoplada por medio de un condensador de 0.1µF para eliminar componentes DC no deseadas, ingresa a la base del transistor,

el cual permite el paso de corriente desde el colector al emisor cada vez que recibe una señal positiva. Esta corriente fluye a través del fotodiodo, generando luz proporcional a la señal original. En este diseño, los capacitores en la entrada y salida del transistor ayudan a reducir el ruido y mejorar la calidad de la señal transmitida ópticamente.

La parte derecha del circuito representa la etapa receptora. Aquí, un fototransistor detecta la luz emitida por el fotodiodo, lo que permite el paso de corriente entre su colector y emisor. Este flujo de corriente produce un voltaje en la resistencia conectada a tierra (1k), que luego es alimentado a un amplificador operacional LM741 en configuración no inversora, para amplificar la señal detectada. Finalmente, la salida del amplificador se dirige al DAQ o microcontrolador para ser procesada digitalmente, donde se puede graficar o analizar su contenido espectral (por ejemplo, mediante FFT como se explicó previamente).

Este sistema optoelectrónico ofrece una forma efectiva y segura de transmitir señales analógicas moduladas con aislamiento eléctrico, especialmente útil en entornos con alta interferencia electromagnética o cuando se requiere evitar conexiones físicas entre etapas electrónicas.

4.3. Capturar señal AM en Python

La siguiente implementación en `Arduino` permite leer una señal analógica proveniente de una fuente externa (como un generador AM desde otro Arduino o un circuito de modulación) y transmitirla al puerto serial hacia un computador. Se toma una muestra del pin analógico `A0` cada milisegundo (equivalente a una frecuencia de muestreo de 1000 Hz), utilizando la función `analogRead`. Esta señal leída, que varía entre 0 y 1023, se envía directamente por el puerto serial como una línea de texto.

Listing 1: Código Arduino: lectura analógica y transmisión serial

```
const int pinEntrada = A0; // Pin donde llega la seal analgica
const unsigned long tasaMuestreo_us = 1000; // 1 ms = 1000 Hz

void setup() {
  Serial.begin(115200); // Velocidad del puerto serial
}

void loop() {
  static unsigned long t_anterior = 0;
  unsigned long t_actual = micros();

  if (t_actual - t_anterior >= tasaMuestreo_us) {
    t_anterior = t_actual;

    int valor = analogRead(pinEntrada); // Lee valor de 0 a 1023
    Serial.println(valor); // Enva por serial
  }
}
```

En el computador, se puede utilizar un script en Python para capturar la señal enviada por el Arduino. El siguiente código establece una conexión serial con el puerto correspondiente, y va leyendo línea por línea los valores enviados por el Arduino. Estos valores, una vez convertidos a voltaje (asumiendo una resolución de 10 bits en un rango de 0 a 5 V), son graficados en el dominio del tiempo. Además, se realiza la Transformada Rápida de Fourier (FFT) para observar la composición espectral de la señal recibida.

Listing 2: Código Python: recepción serial, graficación y FFT

```
import numpy as np
import matplotlib.pyplot as plt
import serial
import time

# Parmetros
fs = 1000 # Frecuencia de muestreo (Hz)
duracion = 2 # Duracin total (segundos)
n_muestras = fs * duracion

# Configura el puerto serial (ajusta el nombre del puerto)
puerto = serial.Serial('COM5', 115200)
time.sleep(2) # Espera para estabilizar conexin

datos_recibidos = []

print("Recibiendo datos...")

while len(datos_recibidos) < n_muestras:
    if puerto.in_waiting > 0:
        linea = puerto.readline().decode('utf-8').strip()
        try:
            valor = int(linea)
            datos_recibidos.append(valor)
        except ValueError:
            continue

puerto.close()
print("Recepcin completa.")

# Procesamiento
senal_adc = np.array(datos_recibidos)
t = np.linspace(0, duracion, len(senal_adc))
senal_voltaje = (senal_adc / 1023) * 5 # Conversin a voltaje

# FFT
fft_senal = np.fft.fft(senal_voltaje)
frecuencias = np.fft.fftfreq(len(t), 1/fs)
n = len(t) // 2

# Grficas
plt.figure(figsize=(12, 6))

plt.subplot(2, 1, 1)
plt.plot(t, senal_voltaje, color='blue')
plt.title('Seal recibida desde Arduino')
plt.xlabel('Tiempo (s)')
plt.ylabel('Voltaje (V)')
plt.grid(True)

plt.subplot(2, 1, 2)
plt.plot(frecuencias[:n], np.abs(fft_senal[:n]), color='purple')
plt.title('FFT de la seal recibida')
```

```
plt.xlabel('Frecuencia␣(Hz)')
plt.ylabel('Magnitud')
plt.grid(True)

plt.tight_layout()
plt.show()
```

5. Elementos a utilizar

1. Fototransistor (receptor)
2. Computador
3. Diodo (emisor).
4. Transistor 2N2222
5. Amplificador lm358
6. Arduino 1
7. Capacitor 0.1 μF
8. Fototransistor

Referencias

[1] M. Banzi and M. Shiloh, *Getting Started with Arduino*, 3rd ed. Sebastopol, CA, USA: Maker Media, 2014.

[2] A. V. Oppenheim and A. S. Willsky, *Signals and Systems*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall, 1997.

[3] S. W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*, 2nd ed. California Technical Publishing, 1999.

[4] W. McKinney, *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2017.

[5] S. Haykin, *Communication Systems*, 5th ed. New York, NY, USA: Wiley, 2009.