



UAT
Universidad Autónoma
de Tamaulipas



Materia:

**DISEÑO ELECTRÓNICO BASADO EN
SISTEMAS EMBEBIDOS**

Alumno:

Posadas Pérez Isaac Sayeg

Paniagua Rico Juan Julian

García Azzúa Jorge Roberto

Grado y grupo:

8°G

Profesor:

Garcia Ruiz Alejandro Humberto

Tarea 5:

**Instrucciones para trabajar con
escritura/lectura de señales analógicas en
arduino**

Instrucciones para Trabajar con Escritura/Lectura de Señales Analógicas en Arduino

Arduino no solo permite trabajar con señales digitales, sino que también ofrece la capacidad de manejar señales analógicas. Estas señales son continuas y pueden tomar cualquier valor dentro de un rango específico, lo que las hace ideales para aplicaciones que requieren medir variables físicas, como la temperatura, la luz o la presión. En esta investigación, abordaremos cómo leer y escribir señales analógicas en Arduino, con ejemplos prácticos y explicaciones detalladas.

1. Introducción a las Señales Analógicas

Las señales analógicas son representaciones continuas de datos que pueden asumir un rango infinito de valores dentro de un límite definido. Por ejemplo, un sensor de temperatura puede proporcionar una salida analógica que varía de 0V a 5V, donde cada valor de voltaje representa una temperatura específica.

En Arduino, los pines analógicos están diseñados para leer estas señales. A diferencia de las señales digitales, que solo pueden ser HIGH o LOW, las señales analógicas se representan mediante un valor en un rango específico, generalmente entre 0 y 1023, debido a la resolución de 10 bits del convertidor analógico-digital (ADC) integrado en las placas Arduino.

2. Lectura de Señales Analógicas

La lectura de señales analógicas en Arduino se realiza utilizando la función `analogRead(pin)`, donde `pin` es el número del pin analógico (A0, A1, A2, etc.). Esta función devuelve un valor entre 0 y 1023, que corresponde al voltaje entre 0V y 5V.

2.1. Ejemplo de Lectura de un Potenciómetro

Un ejemplo común es leer el valor de un potenciómetro, un componente que permite variar la resistencia y, por lo tanto, el voltaje de salida. A continuación se muestra un

```
void setup() {  
    Serial.begin(9600); // Inicia la comunicación serie a 9600 bps  
}  
  
void loop() {  
    int valor = analogRead(A0); // Lee el valor del potenciómetro  
    Serial.println(valor);      // Imprime el valor en el monitor serie  
    delay(500);                // Espera medio segundo  
}
```

3. Escritura de Señales Analógicas

Aunque Arduino no puede generar señales analógicas continuas, puede simular una salida analógica utilizando modulación de ancho de pulso (PWM) a través de pines digitales. La función `analogWrite(pin, value)` se utiliza para establecer el ciclo de trabajo de una señal PWM. El valor puede estar entre 0 (0V) y 255 (5V).

3.1. Ejemplo de Control de Brillo de un LED

A continuación se muestra un ejemplo que utiliza un potenciómetro para controlar el brillo de un LED conectado al pin 9. El valor leído del potenciómetro se convierte a un rango adecuado para la función `analogWrite`.

```
void setup() {  
    pinMode(9, OUTPUT); // Configura el pin 9 como salida
```

```
}  
  
void loop() {  
    int valor = analogRead(A0);      // Lee el valor del potenciómetro  
    int brillo = map(valor, 0, 1023, 0, 255); // Mapea el valor a un rango de 0 a 255  
    analogWrite(9, brillo);          // Establece el brillo del LED  
    delay(10);                       // Espera un corto tiempo  
}
```

4. Consideraciones al Trabajar con Señales Analógicas

4.1. Resolución del ADC

El convertidor analógico-digital (ADC) de Arduino tiene una resolución de 10 bits, lo que significa que puede representar 1024 niveles diferentes (desde 0 hasta 1023). Esto determina la precisión de las lecturas analógicas. En aplicaciones que requieren mayor precisión, se pueden considerar placas Arduino con resoluciones superiores, como Arduino Due, que cuenta con un ADC de 12 bits.

4.2. Ruido en las Lecturas

Las lecturas analógicas pueden verse afectadas por el ruido eléctrico y otros factores. Para obtener lecturas más estables, se pueden implementar técnicas de suavizado, como promediar múltiples lecturas en lugar de utilizar una sola.

```
int leerPromedio(int pin) {  
    int suma = 0;  
    for (int i = 0; i < 10; i++) {
```

```
    delay(10);           // Espera entre lecturas  
  }  
  return suma / 10; // Retorna el promedio  
}
```

4.3. Conexión de Sensores

Al conectar sensores analógicos, como sensores de temperatura o de luz, es importante tener en cuenta la alimentación y las características de salida del sensor. Algunos sensores requieren circuitos adicionales, como divisores de tensión o filtros, para obtener lecturas precisas.

5. Conclusiones

Trabajar con señales analógicas en Arduino es necesario para medir y controlar variables físicas en proyectos electrónicos. La capacidad de leer y simular señales analógicas permite a los usuarios desarrollar aplicaciones interactivas y responder a condiciones del entorno. Comprender cómo utilizar las funciones `analogRead` y `analogWrite`, así como las consideraciones importantes relacionadas con la resolución y el ruido, es esencial para el éxito en proyectos de Arduino.

Bibliografía

1. Banzi, M., & Shiloh, M. (2014). *Getting Started with Arduino*. Maker Media, Inc.
 - Este libro proporciona una introducción a la plataforma Arduino, que incluye ejemplos de lectura y escritura de señales analógicas.
2. Monk, S. (2015). *Programming Arduino: Getting Started with Sketches*. McGraw-Hill Education.
 - Un recurso que presenta ejemplos prácticos sobre la programación en Arduino, incluidas señales analógicas.

- Documentación oficial de Arduino que detalla las funciones para trabajar con señales analógicas.
4. Adafruit. (n.d.). *Learning System*. Retrieved from <https://learn.adafruit.com/>
- Un recurso en línea que ofrece tutoriales sobre programación de Arduino y ejemplos prácticos.
5. Simon Monk. (2014). *Arduino Cookbook*. O'Reilly Media.
- Un libro que ofrece una colección de recetas y ejemplos prácticos para trabajar con Arduino en diferentes proyectos.