

INSTITUTO TECNOLÓGICO DE SALTILLO

REPORTE DE PRACTICA 3: LECTURA ANALOGA

Presenta:

Kevin Alejandro Fuentes Martínez

Orlando Ávila Céspedes

Eduardo Ernesto Castillo Cordero

Rogelio Pérez Guevara

Materia:

Sistemas Programables

Profesor:

ING. Mona Peña Luis Javier

07/12/2024

Contenido

Objetivo General	3
Marco Teórico	3
Desarrollo de la Práctica	3
Materiales y Equipos	3
Procedimiento	4
Programación del Arduino	4
Explicaciones del Código	5
Declaración de pines:	5
Lectura de la entrada analógica:	5
Lógica para LEDs:	6
Optimización de la salida serial:	6
Imágenes del Circuito	7
Conclusiones	8

Objetivo General

Desarrollar un sistema que permita monitorear y visualizar el nivel de voltaje de una señal analógica a través de LEDs, utilizando un microcontrolador Arduino. Este sistema busca representar estados de carga (cargado, medio y mínimo) de forma visual para aplicaciones prácticas como monitoreo de baterías.

Marco Teórico

El uso de Arduino en la electrónica permite la interacción con señales analógicas y digitales para múltiples aplicaciones.

1. **Entradas analógicas:** Estas señales son leídas por el conversor analógico-digital (ADC) del Arduino y convertidas a valores digitales entre 0 y 1023, dependiendo del nivel de voltaje.
2. **Salidas digitales:** Los pines digitales del Arduino controlan dispositivos externos, como LEDs, en estados HIGH (encendido) o LOW (apagado).
3. **Voltaje:** Se define como la diferencia de potencial eléctrico, medido en voltios (V). El programa convierte las lecturas analógicas en valores de voltaje reales para la interpretación.

En este proyecto, los LEDs representan los estados de un sistema de monitoreo de voltaje:

- **Verde:** Nivel de voltaje óptimo.
- **Ámbar:** Nivel medio de voltaje.
- **Rojo:** Voltaje bajo.

Desarrollo de la Práctica

Se diseñó un circuito basado en Arduino que mide el nivel de voltaje en una entrada analógica y utiliza LEDs para mostrar visualmente el estado del voltaje. Los cambios significativos de voltaje se imprimen en el monitor serial.

Materiales y Equipos

- Arduino Uno.
- 3 LEDs (verde, amarillo, rojo).
- 3 resistencias de 220 Ω .
- Protoboard.
- Cables de conexión.
- Fuente de alimentación para generar señal analógica en el pin.

Procedimiento

1. Ensamblar el circuito en la protoboard:
 - Conectar los LEDs con sus respectivas resistencias a los pines digitales 4, 5 y 6 del Arduino.
 - Conectar una señal analógica al pin .
2. Programar el Arduino con el código proporcionado.
3. Subir el código al Arduino utilizando el IDE de Arduino.
4. Alimentar el circuito y observar los estados de los LEDs según el nivel de voltaje.
5. Monitorear las lecturas de voltaje en el monitor serial del IDE.

Programación del Arduino

El código utiliza funciones básicas del Arduino para leer y procesar señales analógicas, y para controlar los LEDs en función del nivel de voltaje.

```
//pines de led
int ledV = 4;
int ledA = 5;
int ledR = 6;

//variables
int valorAng = 0;
float voltaje = 0;
float voltajeAnterior =0;

//niveles
float cargado = 1.5;
float medio = 1.3;
float minimo = 0.3;

void setup(){
  Serial.begin(9600);
  pinMode(ledV,OUTPUT);
  pinMode(ledA,OUTPUT);
  pinMode(ledR,OUTPUT);
}

void loop(){
  //Leemos entrada analogica
  valorAng = analogRead(A0);
  //obtenemos el voltaje
  voltaje = 0.0048 * valorAng;
  //Serial.print("Voltaje: " + voltaje);
```

```

// Solo imprimimos si hay un cambio significativo en el voltaje
if (abs(voltaje - voltajeAnterior) > 0.01) {
    Serial.print("Voltaje: ");
    Serial.println(voltaje);
    voltajeAnterior = voltaje; // Actualizamos el valor anterior
}

if(voltaje >= cargado){
    digitalWrite(ledV,HIGH);
    delay(1000);
}else if(voltaje < cargado && voltaje >= medio){
    digitalWrite(ledA,HIGH);
    delay(1000);
}else if(voltaje < medio && voltaje >= minimo){
    digitalWrite(ledR,HIGH);
    delay(1000);
}else{
    digitalWrite(ledR,LOW);
    digitalWrite(ledA,LOW);
    digitalWrite(ledV,LOW);
}

delay(500);
}

```

Explicaciones del Código

Declaración de pines:

```

//pines de led
int ledV = 4;
int ledA = 5;
int ledR = 6;

```

Se asignan pines digitales para controlar los LEDs.

Lectura de la entrada analógica:

```

//Leemos entrada analogica
valorAng = analogRead(A0);
//obtenemos el voltaje
voltaje = 0.0048 * valorAng;
//Serial.print("Voltaje: " + voltaje);

```

El ADC convierte la señal analógica en un valor digital, que se escala para obtener el voltaje real.

Lógica para LEDs:

Según el voltaje, se encienden los LEDs correspondientes:

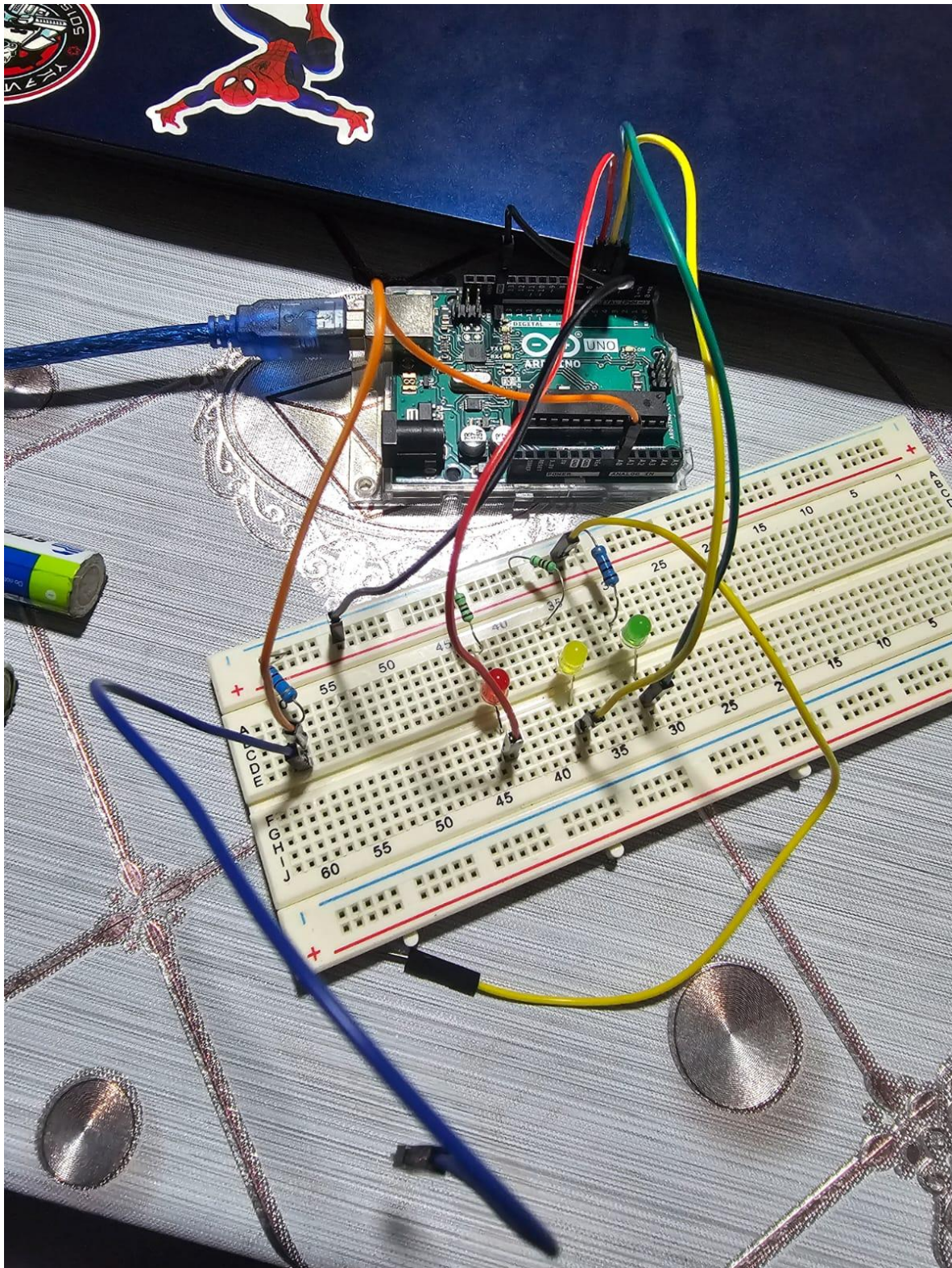
- **Verde:** $\text{Voltaje} \geq 1.5\text{V}$.
- **Amarillo:** $1.3\text{V} \leq \text{Voltaje} < 1.5\text{V}$.
- **Rojo:** $0.3\text{V} \leq \text{Voltaje} < 1.3\text{V}$.
- **Apagado:** $\text{Voltaje} < 0.3\text{V}$.

Optimización de la salida serial:

```
// Solo imprimimos si hay un cambio significativo en el voltaje
if (abs(voltaje - voltajeAnterior) > 0.01) {
    Serial.print("Voltaje: ");
    Serial.println(voltaje);
    voltajeAnterior = voltaje; // Actualizamos el valor anterior
}
```

Se imprime el voltaje solo si cambia significativamente.

Imágenes del Circuito



Conclusiones

El sistema implementado demostró ser efectivo para monitorear niveles de voltaje y proporcionar una representación visual clara de su estado mediante LEDs. Este proyecto puede extenderse para aplicaciones más complejas, como monitoreo de baterías en tiempo real o sistemas de advertencia de sobrecarga. Además, se comprobó la utilidad del Arduino como herramienta versátil en proyectos de electrónica y control.