

**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

**Unidad 3 – Proyecto Final:**

Documentación

## Documentación del Código de Control de Focos

### Propósito

Controla la cantidad de focos encendidos (1, 2 o 3) en función de los niveles de oscuridad detectados por el fototransistor, usando umbrales predefinidos y filtrado por mediana.

### Estructura de Umbrales

const int umbral1 = 700; // Oscuridad mínima: 1 foco

const int umbral2 = 800; // Oscuridad media: 2 focos

const int umbral3 = 900; // Oscuridad máxima: 3 focos

Umbral: umbral3

Valor: 900

Acción: Máxima oscuridad

Umbral: umbral2

Valor: 800

Acción: Oscuridad media

Umbral: umbral1

Valor: 700

Acción: Oscuridad mínima

### Lógica de Control

```
if (mediana >= umbral3) {  
  digitalWrite(ledPin1, HIGH);  
  digitalWrite(ledPin2, HIGH);  
  digitalWrite(ledPin3, HIGH);  
  Serial.println("Oscuridad ALTA - 3 focos encendidos");  
}  
  
else if (mediana >= umbral2) {  
  digitalWrite(ledPin1, HIGH);  
  digitalWrite(ledPin2, HIGH);  
  digitalWrite(ledPin3, LOW);  
}
```

```
Serial.println("Oscuridad MEDIA - 2 focos encendidos");  
}  
else if (mediana >= umbral1) {  
digitalWrite(ledPin1, HIGH);  
digitalWrite(ledPin2, LOW);  
digitalWrite(ledPin3, LOW);  
Serial.println("Oscuridad BAJA - 1 foco encendido");  
}  
else {  
digitalWrite(ledPin1, LOW);  
digitalWrite(ledPin2, LOW);  
digitalWrite(ledPin3, LOW);  
Serial.println("Luz suficiente - Todos los focos apagados");  
}
```

## Flujo de Funcionamiento

### 1. Comparación Jerárquica:

- Evalúa primero la condición más crítica ( $\text{mediana} \geq \text{umbral3}$ ).
- Si no se cumple, verifica el siguiente nivel ( $\text{mediana} \geq \text{umbral2}$ ), y así sucesivamente.

### 2. Salidas Digitales:

- Los pines 8, 9 y 10 se activan/desactivan según corresponda:
  - HIGH: Foco encendido.
  - LOW: Foco apagado.

### 3. Retroalimentación por Serial:

- Envía mensajes descriptivos al monitor serial para depuración.
  - Ejemplo: "Oscuridad MEDIA - 2 focos encendidos".

## Relación entre Valores y Oscuridad

Condición: Mediana  $\geq 900$

Comportamiento: 3 focos encendidos (máxima oscuridad)

Valores de Ejemplo: 900-1023

Condición: Mediana  $\geq 800$

Comportamiento: 2 focos encendidos

Valores de Ejemplo: 800-899

Condición: Mediana  $\geq 700$

Comportamiento: 1 foco encendido

Valores de Ejemplo: 700-799

Condición: Mediana  $< 700$

Comportamiento: Todos los focos apagados

Valores de Ejemplo: 0-699

## Ejemplo de Salida en Monitor Serial

Oscuridad ALTA - 3 focos encendidos

Mediana: 927 | Valor crudo: 935

Oscuridad MEDIA - 2 focos encendidos

Mediana: 815 | Valor crudo: 803

Luz suficiente - Todos los focos apagados

Mediana: 432 | Valor crudo: 445

## Cableado

### Componentes Clave

1. **Arduino:** Controla todo.

2. **Fototransistor:** Mide la luz (conectado a un pin analógico como **A0**).
3. **3 Relevadores:** Cada uno conectado a los pines **8, 9 y 10** del Arduino.
4. **Batería de 9V:** Alimenta los focos de 6V.
5. **Focos de 6V:** Encienden/apagan según la oscuridad.

#### 1. **Fototransistor:**

- El pin largo va al +5V del Arduino.
- El pin corto va a A0 y también a una resistencia que conectas a GND (esto ya lo tienes).

#### 2. **Relevadores:**

- Cada relevador tiene 3 partes importantes:
  - Señal de control: Conectada a los pines 8, 9 y 10 del Arduino.
  - Entrada de energía: Conectada a la batería de 9V (polo positivo).
  - Salida a foco: Conectada al foco de 6V.

#### 3. **Conexiones de los Focos:**

- El polo positivo de cada foco va al relevador correspondiente (pin 8 → foco 1, pin 9 → foco 2, etc.).
- El polo negativo de todos los focos va al polo negativo de la batería de 9V.

## Funcion general

#### 1. **Fototransistor:**

- Si hay oscuridad, envía un valor alto al Arduino (ej. 900).
- Si hay luz, envía un valor bajo (ej. 100).

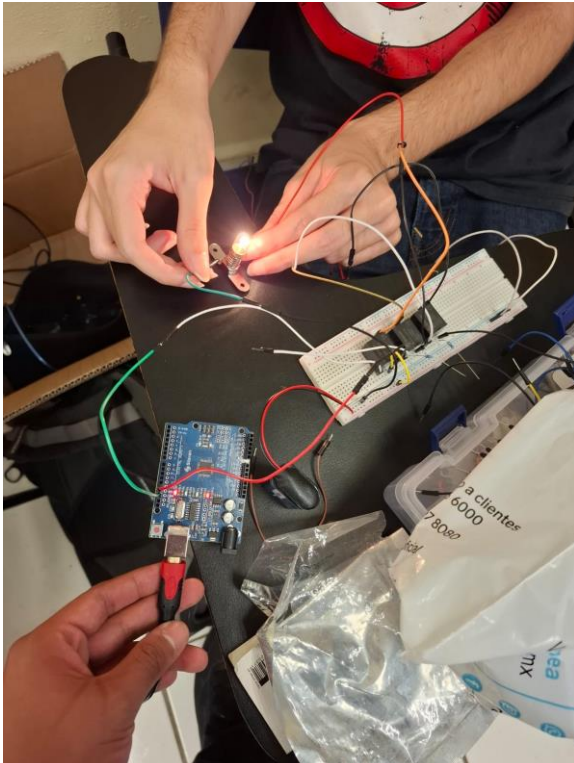
## 2. Arduino:

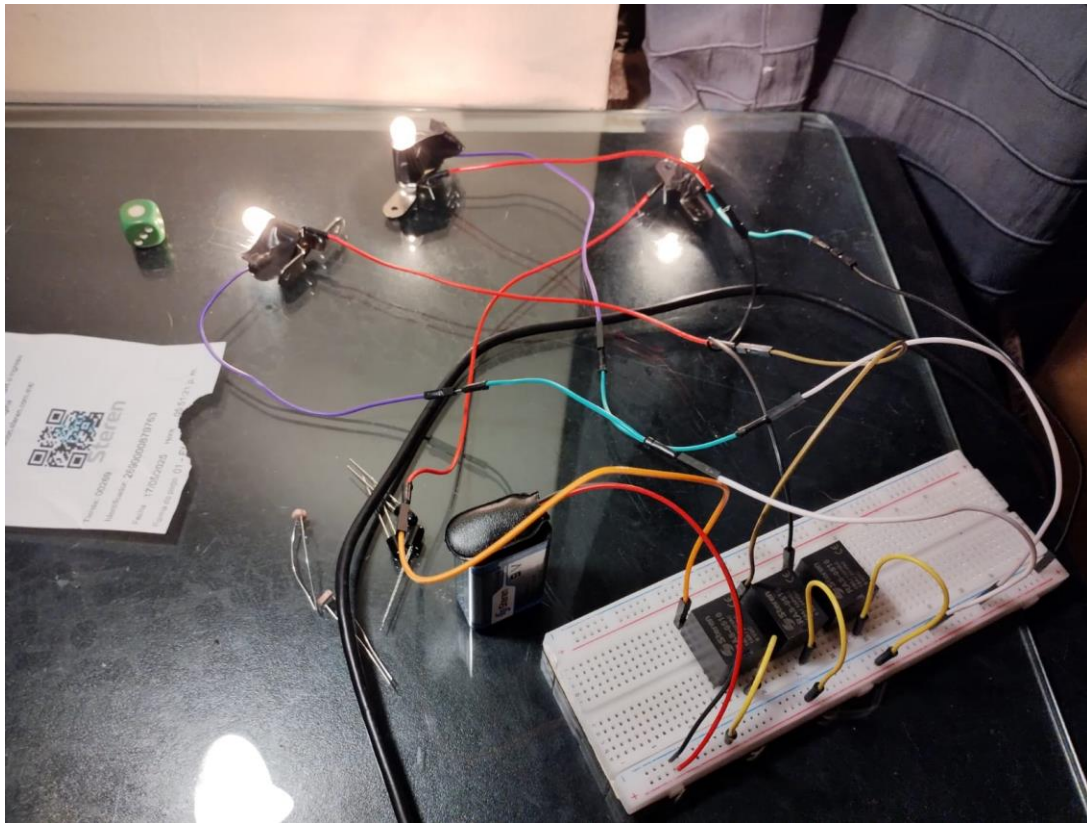
- Usando el código que ya tiene:
  - Si hay mucha oscuridad (valor alto), activa los pines 8, 9 y 10 → enciende los 3 focos.
  - Si hay oscuridad media, activa los pines 8 y 9 → enciende 2 focos.
  - Si hay poca oscuridad, activa solo el pin 8 → enciende 1 foco.
  - Si hay luz suficiente, no activa ningún pin → todos los focos apagados.

## 3. Relevadores:

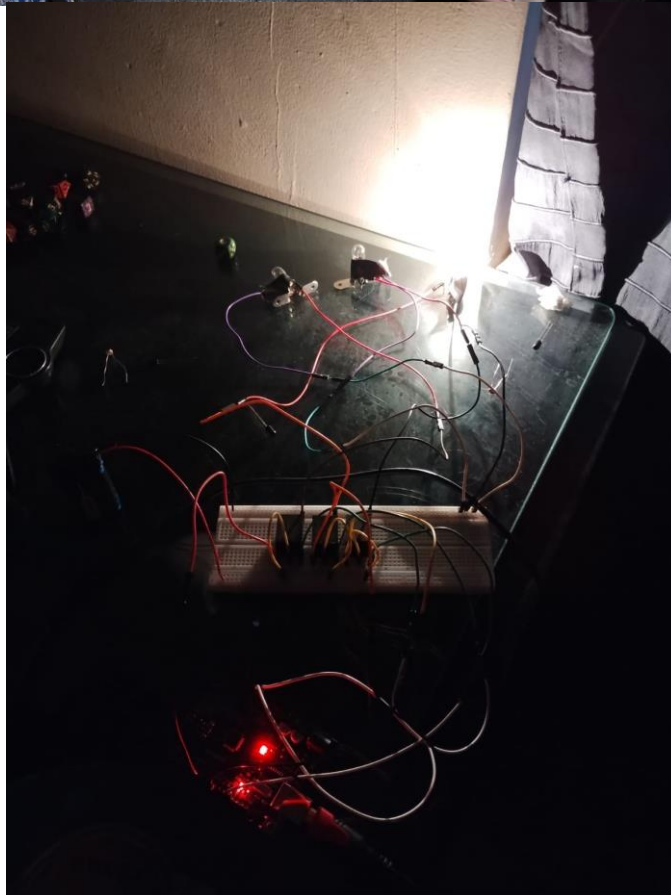
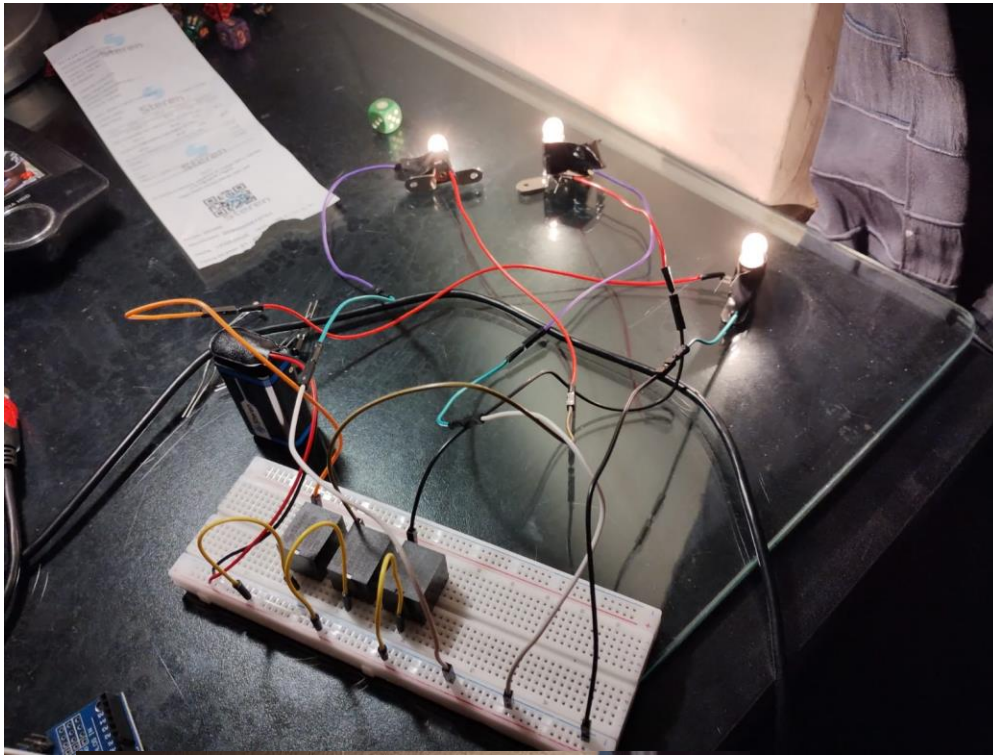
- Cuando un pin del Arduino (ej. pin 8) se activa, el relevador correspondiente cierra el circuito, permitiendo que la batería de 9V envíe energía al foco de 6V.

## Capturas:









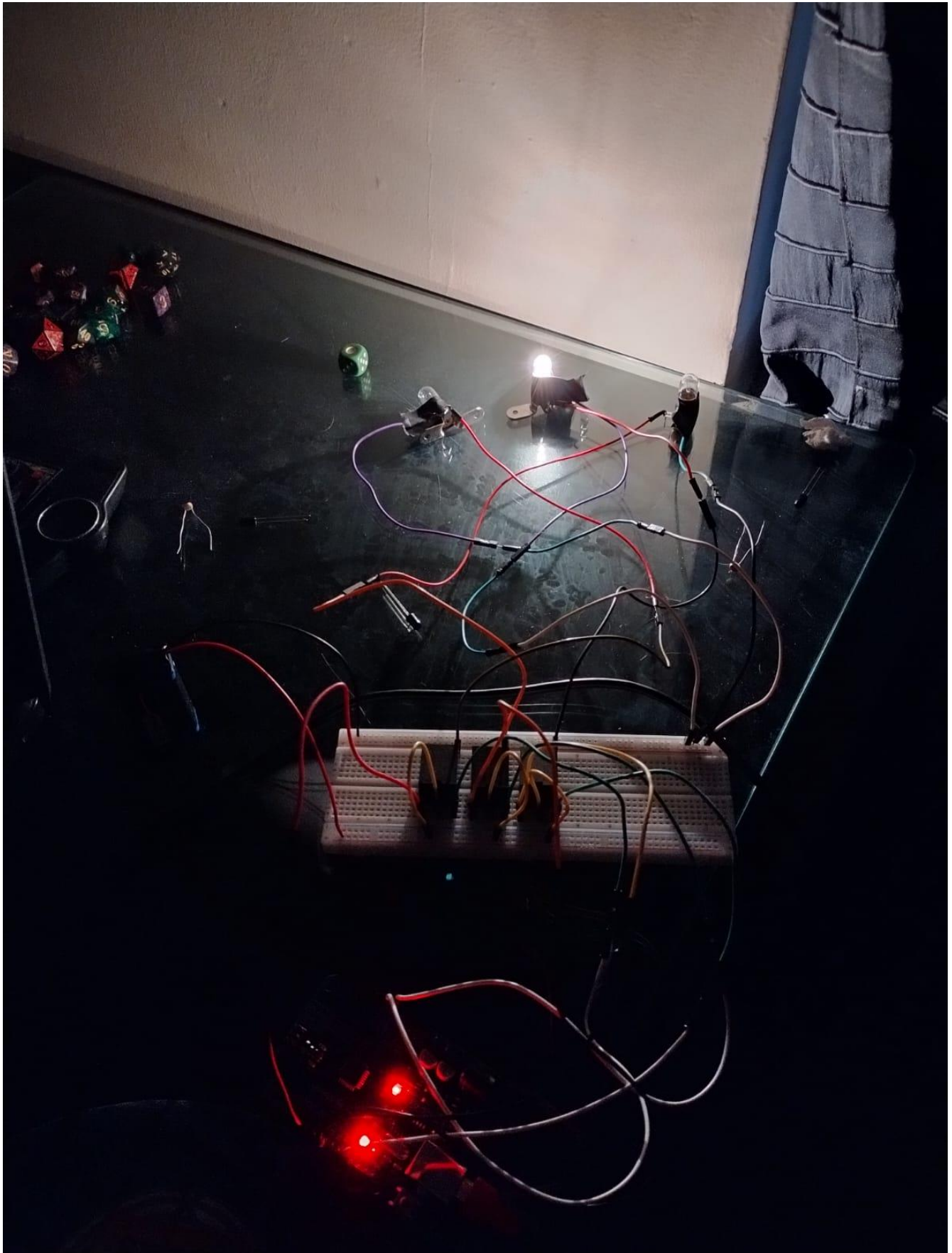


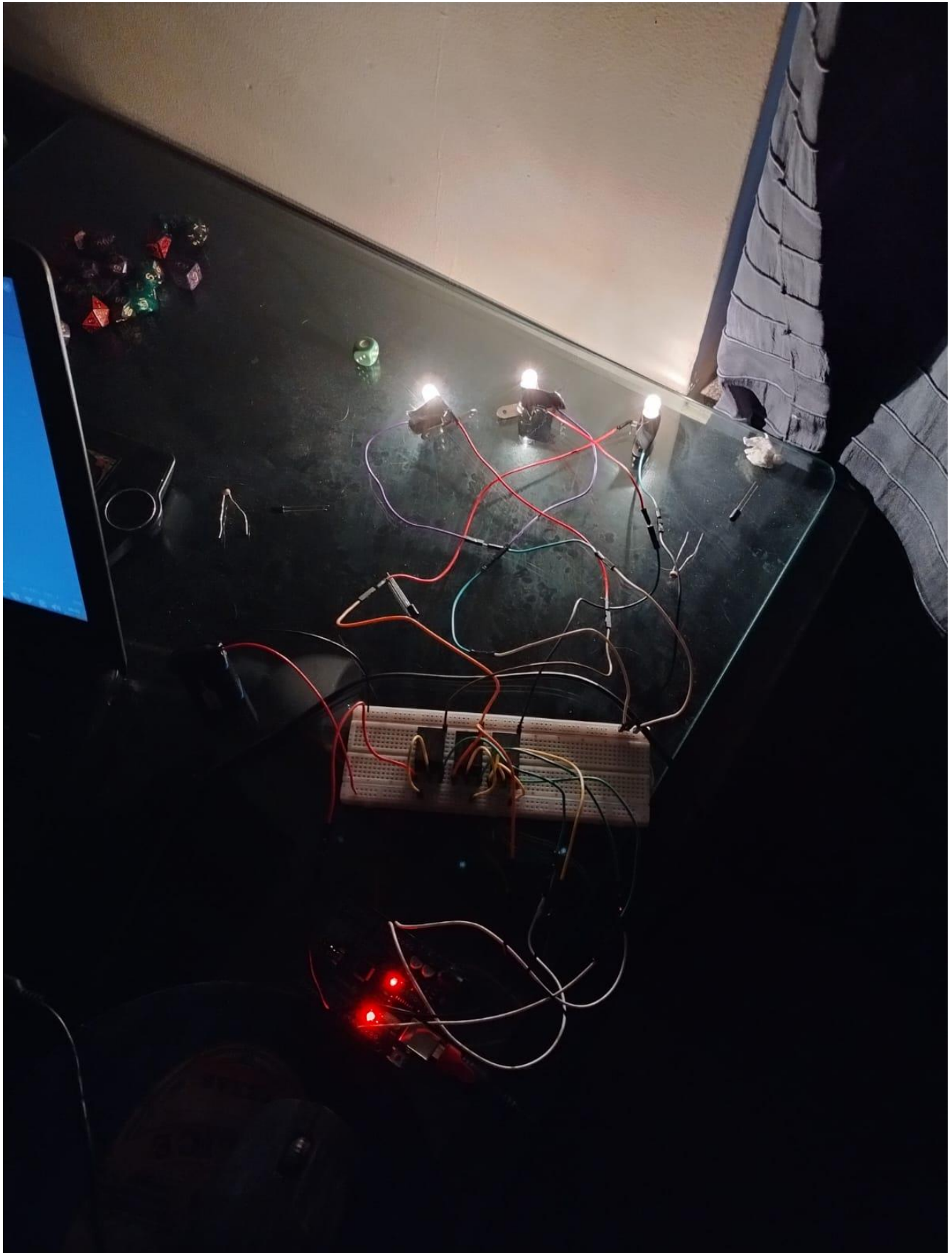


**UAT**  
Universidad Autónoma  
de Tamaulipas



Facultad de Ingeniería  
Tampico





## Codigo completo:

```
const int sensorPin = A0;    // Pin del fototransistor

const int ledPin1 = 8;       // Foco 1
const int ledPin2 = 9;       // Foco 2
const int ledPin3 = 10;      // Foco 3


// Umbrales de oscuridad

const int umbral1 = 700;     // 1 foco oscuro
const int umbral2 = 800;     // 2 focos
const int umbral3 = 900;     // 3 focos


// Variables para el filtrado

const int numLecturas = 5;   // Número de lecturas para la mediana
int lecturas[numLecturas];   // Array para almacenar lecturas
int indice = 0;              // Índice actual


void setup() {
    Serial.begin(9600);

    pinMode(ledPin1, OUTPUT);
    pinMode(ledPin2, OUTPUT);
    pinMode(ledPin3, OUTPUT);
}
```

```
void loop() {  
  
    // Lee el sensor y actualiza el array de lecturas  
  
    lecturas[indice] = analogRead(sensorPin);  
  
    indice = (indice + 1) % numLecturas;  
  
  
    // Calcula la mediana de las últimas lecturas  
  
    int mediana = calcularMediana(lecturas, numLecturas);  
  
  
    // Control de focos según la mediana  
  
    if (mediana >= umbral3) {  
  
        digitalWrite(ledPin1, HIGH);  
  
        digitalWrite(ledPin2, HIGH);  
  
        digitalWrite(ledPin3, HIGH);  
  
        Serial.println("Oscuridad ALTA - 3 focos encendidos");  
  
    }  
  
    else if (mediana >= umbral2) {  
  
        digitalWrite(ledPin1, HIGH);  
  
        digitalWrite(ledPin2, HIGH);  
  
        digitalWrite(ledPin3, LOW);  
  
        Serial.println("Oscuridad MEDIA - 2 focos encendidos");  
  
    }  
  
    else if (mediana >= umbral1) {  
  
        digitalWrite(ledPin1, HIGH);  
  

```

```
digitalWrite(ledPin2, LOW);  
digitalWrite(ledPin3, LOW);  
Serial.println("Oscuridad BAJA - 1 foco encendido");  
}  
else {  
    digitalWrite(ledPin1, LOW);  
    digitalWrite(ledPin2, LOW);  
    digitalWrite(ledPin3, LOW);  
    Serial.println("Luz suficiente - Todos los focos apagados");  
}  
  
// Muestra valores en el monitor serial  
Serial.print("Mediana: ");  
Serial.print(media);  
Serial.print(" | Valor crudo: ");  
Serial.println(analogRead(sensorPin));  
  
delay(500); // Espera entre lecturas  
}  
  
// Función para calcular la mediana (ordena y toma el valor central)  
int calcularMediana(int* array, int n) {  
    // Copia el array para no modificar el original  
    int copia[n];
```



```
for (int i = 0; i < n; i++) copia[i] = array[i];
```

```
// Ordena la copia (método burbuja simple)
```

```
for (int i = 0; i < n - 1; i++) {  
    for (int j = 0; j < n - i - 1; j++) {  
        if (copia[j] > copia[j + 1]) {  
            int temp = copia[j];  
            copia[j] = copia[j + 1];  
            copia[j + 1] = temp;  
        }  
    }  
}
```

```
// Devuelve la mediana
```

```
return copia[n / 2];  
}
```