

# Reporte

## Prácticas de laboratorio

---



## Introducción

Durante este primer parcial se trabajó con el microcontrolador Arduino Uno y nos enfocamos en aprender todas las opciones que este ofrece, tales como

- Interrupciones
- Timers
- Señales de tipo:
  - Analógicas
  - Digitales
  - PWM

Además, aprendimos a trabajar con distintos componentes electrónicos, como:

- LEDs
- Resistencias
- Potenciómetros
- Switches pulsadores
- Sensores IR
- Transistores
- Fuentes de energía.

Todos estos conceptos fueron vistos realizando diversas prácticas durante el parcial, además de una práctica final, la cual comprende todo lo visto en clase.

Adicionalmente, en este reporte incluimos la descripción de la práctica correspondiente, una imagen del circuito como evidencia de la realización de la práctica, el código utilizado para programar el Arduino Uno y una hoja de datos que muestra la conexión de cada uno de los circuitos utilizados.

# Desarrollo

## Practica 1

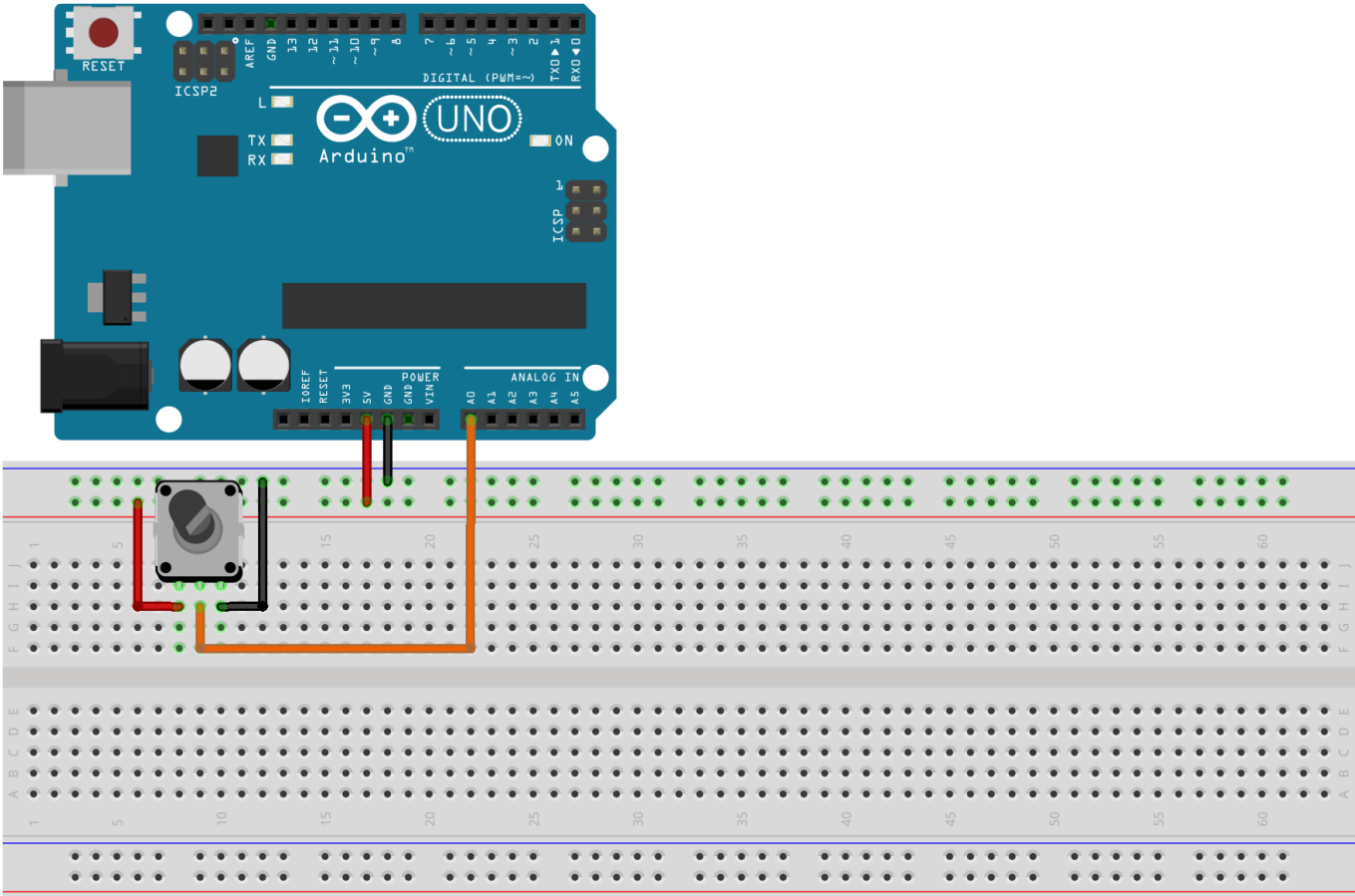
---

Durante esta práctica, con ayuda del Arduino buscamos comprender el funcionamiento de un potenciómetro, así como la utilidad de los puertos análogos del Arduino y las instrucciones que los acompañan. Para ello, conectamos voltaje y tierra en cada extremo del potenciómetro (tal y como haríamos con una resistencia normal) y del pin medio obtenemos la señal que mandamos al puerto "A0" del Arduino, y que es leído con la instrucción "analogRead".

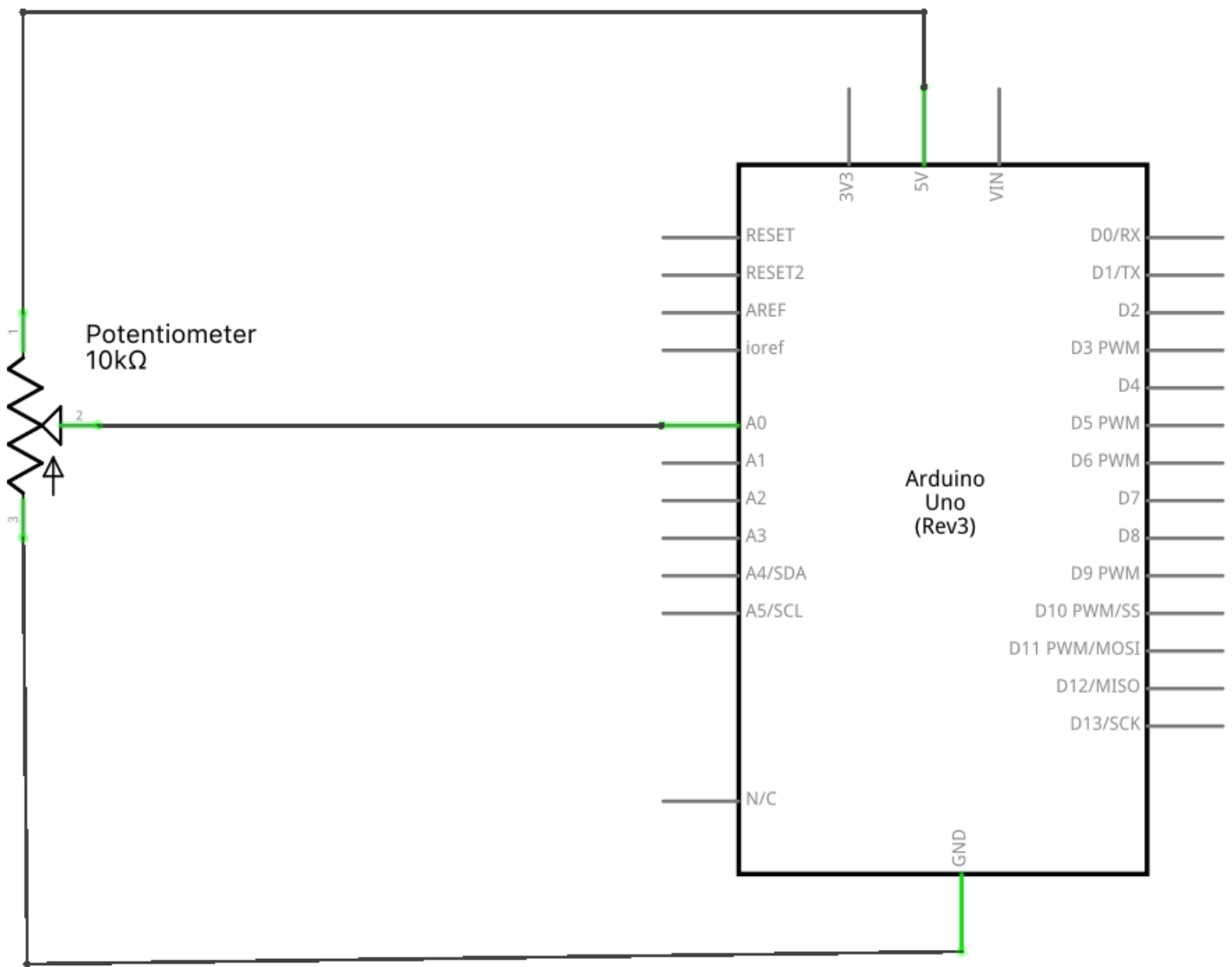
A continuación se muestra el código utilizado para la práctica:

```
void setup() {  
    Serial.begin(9600);  
}  
  
void loop() {  
    int value = analogRead(A0);  
    Serial.print("Valor A0: ");  
    Serial.println(value);  
    delay(1);  
}
```

A continuación se muestra el diagrama de conexión del circuito:

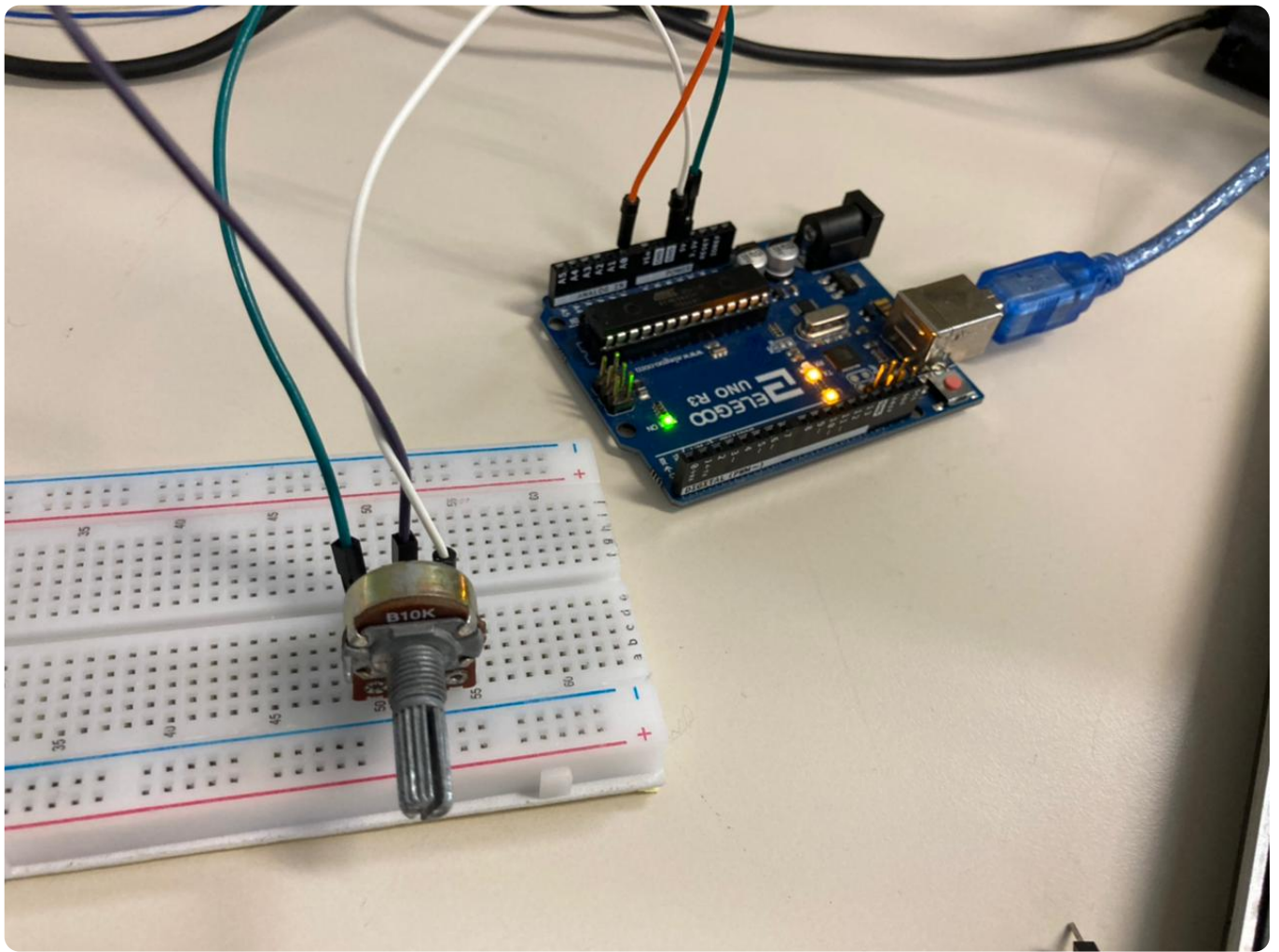


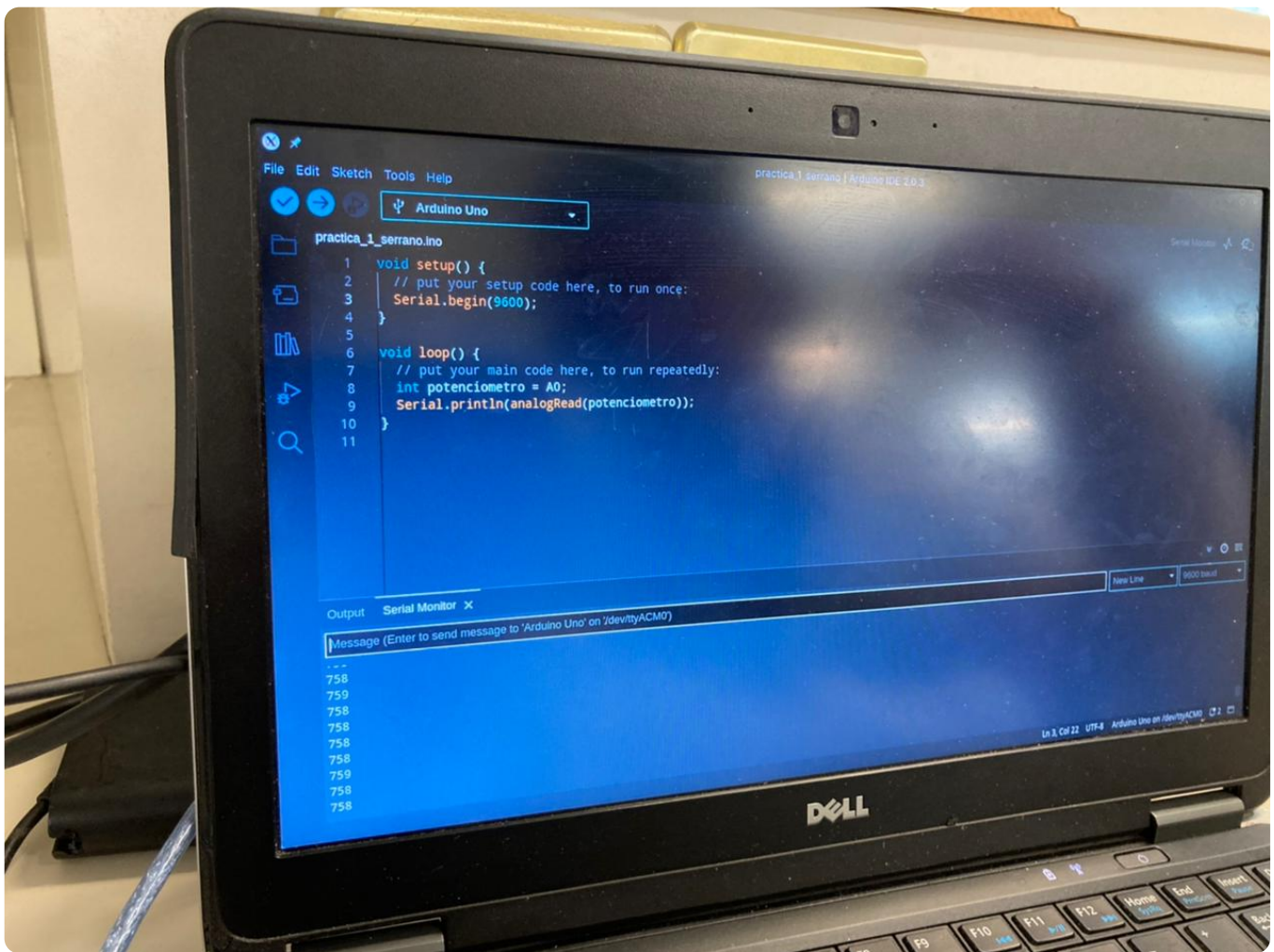
fritzing



fritzing

Finalmente se incluye la foto de evidencia del laboratorio:





## Practica 2 – Delay con potenciómetro

Para esta practica aplicaremos el conocimiento obtenido durante la práctica anterior para controlar los intervalos de prendido y apagado de un foco led intermitente, esto mandando la lectura del potenciómetro directamente a la función “delay” que viene incorporada en el Arduino, además de incorporar la función “digitalWrite” para prender y apagar el foco led según si le damos el valor “HIGH” o “LOW” en el código.

```
void setup() {  
    pinMode(9, OUTPUT);  
    Serial.begin(9600);  
}  
  
void loop() {  
    Serial.println(analogRead(A0));  
  
    digitalWrite(9, HIGH);  
    delay(analogRead(A0));  
  
    digitalWrite(9, LOW);  
    delay(analogRead(A0));  
}
```

```
}
```

### Practica 3 – Control de voltaje con potenciómetro

Al igual que con la practica anterior, en esta ocasión utilizaremos los conceptos aprendidos en la primera práctica, sin embargo, ahora se utiliza el potenciómetro para calcular una relación entre su propio valor y el de la resistencia, es decir que, digamos si el valor máximo del potenciómetro es igual a 5 volts, según una medida 'x' ¿cuánto voltaje obtenemos?, se puede ver que es una regla de tres, y así mismo viene incluido en el código.

### Practica 4 – PWM

Con esta practica buscamos comprender el funcionamiento y la utilidad de los pulsos PWM al aplicarlos sobre un foco led. Para el funcionamiento entendemos que, como el nombre indica los "pulsos PWM" son claramente pulsos, es decir, subidas y bajadas rápidas de voltaje en intervalos definidos. En este caso utilizamos los pulsos para implementar un foco led que se enciende y apaga de forma gradual en lugar de cambiarse instantáneamente.