Workshop: Analog input with a Potentiometer

Figure 1: Différents Potentiomètres

In this workshop, we will read the analog value of a potentiometer using an Arduino.

We will explore how analog values can be used to control other electronic components.

Required Materials

- 1x Arduino Uno board
- 1x Potentiometer
- Jumper wires
- Breadboard (optional)
- USB cable for Arduino

Part 1: Set Up the Circuit

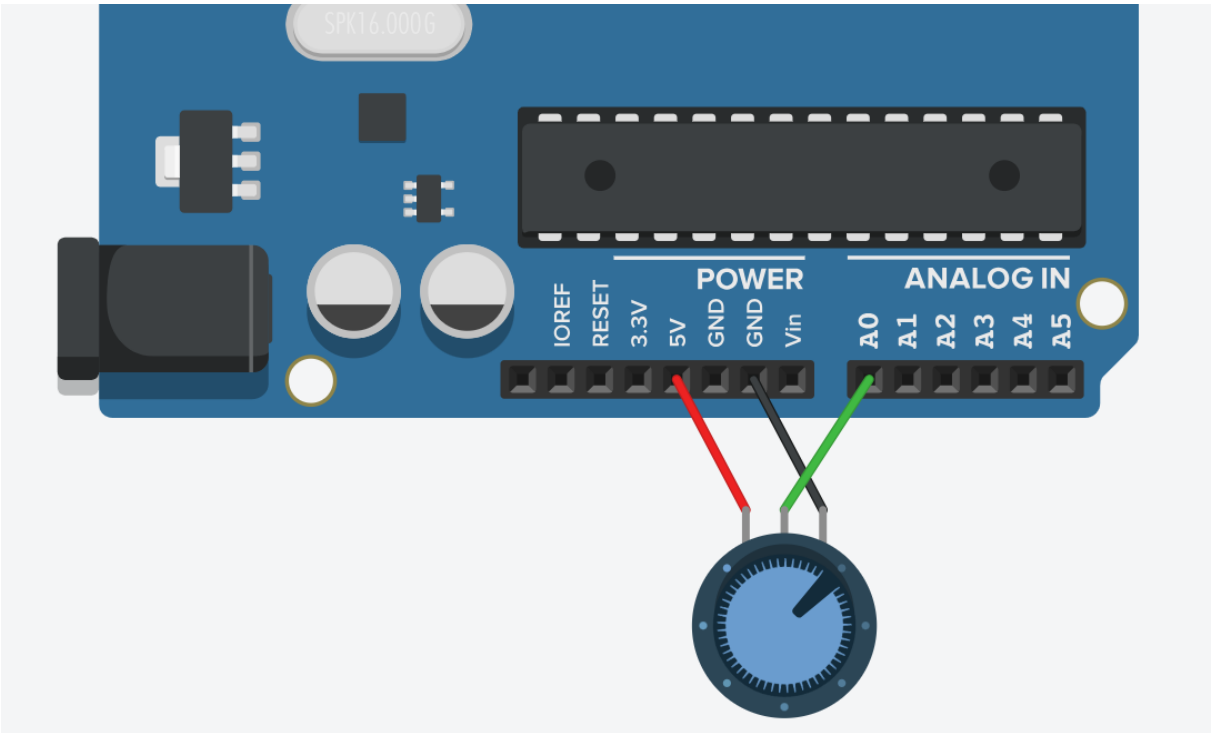

Figure 2: Potentiometer Circuit Setup

- **Pin 1 (Power):** Connect it to the 5V pin on the Arduino.
- **Pin 2 (Signal):** Connect it to an analog pin (e.g., `A0`) on the Arduino.
- **Pin 3 (Ground):** Connect it to a GND pin on the Arduino.

Part 2: Program the Arduino

Open the Arduino IDE.

Step 1: Declare the Analog Pin

Start by declaring a macro for the potentiometer pin, which acts as an alias for the analog pin used (`A0`):

```
#define POT_PIN A0
```

Step 2: Set Up Serial Monitor

To communicate via the serial port, initialize it in the `setup()` function:

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

Step 3: Read the Analog Value

In the `loop()` function, read the potentiometer value and display it:

```
int value;

void loop() {
   value = analogRead(POT_PIN); // Read & store the analog value

   Serial.println(value);       // Display the value
   delay(500);                  // Wait for 500 milliseconds
}
```

With this code, the `loop()` function will continuously read the potentiometer value and print it to the serial monitor every 500 milliseconds.

Part 3: Understanding and Exploration

Questions to Explore

1. Observe the value displayed on the Serial Monitor. What happens when you turn the potentiometer?
2. How do the values read change when the potentiometer is turned completely in one direction versus the other?
3. How could you use these values to control a servo motor?