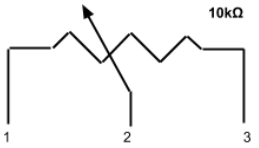
Basic potentiometer use on an Arduino

A potentiometer is a variable resistor with three attachment points. We can take a look at a schematic drawing of the device to better understand what's going on inside:



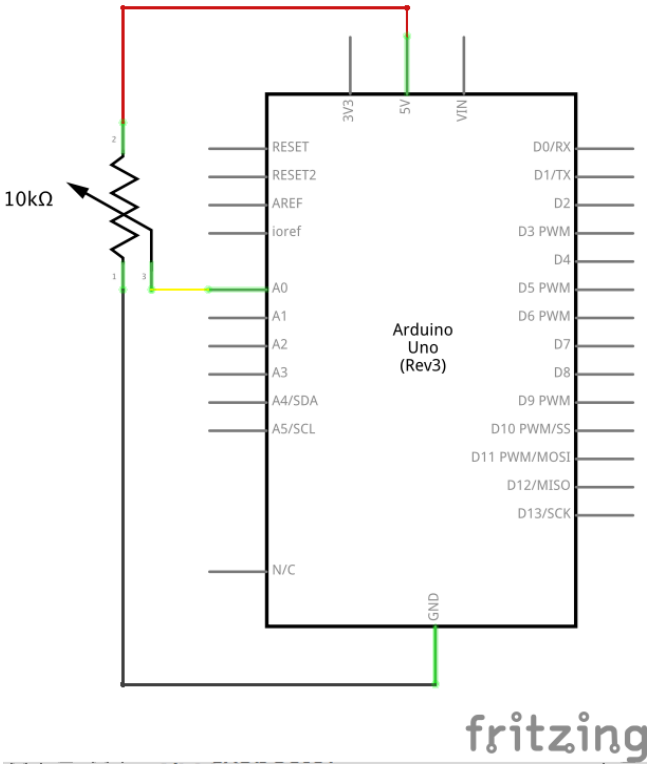
There are two outside legs, here numbered 1 and 3. These are connected to each other by a resistor, whose value is written near the potentiometer: in this case, it's 10kΩ (10,000 Ohms). This resistance doesn't change.

If you only used pins 1 and 3, this potentiometer would simply be a 10kΩ resistor. It's the last leg, #2 in this diagram, where the device actually does something. Leg #2 is called the *wiper*, and it can be moved physically (by turning a knob or sliding a slider) so that it electrically attaches to the resistor inside the potentiometer in different locations.

When this wiper is all the way to the left, there is no resistance between it and pin 1: in this case, pins 1 and 2 have 0Ω between them, and pins 2 and 3 have 10kΩ between them—vice versa if it's turned to the right.

The real magic happens in the middle: if the wiper were, for instance, perfectly centered, then there would be 5kΩ of resistance between pins 1 and 2, and also 5kΩ of resistance between pins 2 and 3. By moving the wiper to any intermediate point between far left and far right, any resistance between 0Ω and 10kΩ can be selected.

The typical Arduino wiring is to attach pin 1 to ground; pin 3 to 5V; and pin 2 (the wiper) to any analog input on the Arduino. Then, performing an `analogRead()` on that pin will produce a value that changes based on the position of the wiper.



Sample Arduino sketch to do `analogRead()` and get `Serial` feedback

```
int POTPIN = A0; // variable to store the pin the potentiometer is wired to

void setup() {
  pinMode(POTPIN, INPUT); // setting the potentiometer pin as an input
  Serial.begin(9600); // starts serial communication at 9,600 baud (the rate)
}

void loop() {
  int readVal; // initialize a new integer to store the potentiometer value
  readVal = analogRead(POTPIN); // do the analog read and store the value
  Serial.println(readVal); // push the most recent value to the computer
  delay(50); // slow the loop down a bit before it repeats
}
```