Analogue Input

In this example we use a variable resistor (a potentiometer or a photoresistor), we read its value using one analogue input of an Arduino and we change the blink rate of the built-in LED accordingly. The resistor's analogue value is read as a voltage because this is how the analogue inputs work.

Hardware Required

1) Arduino Board

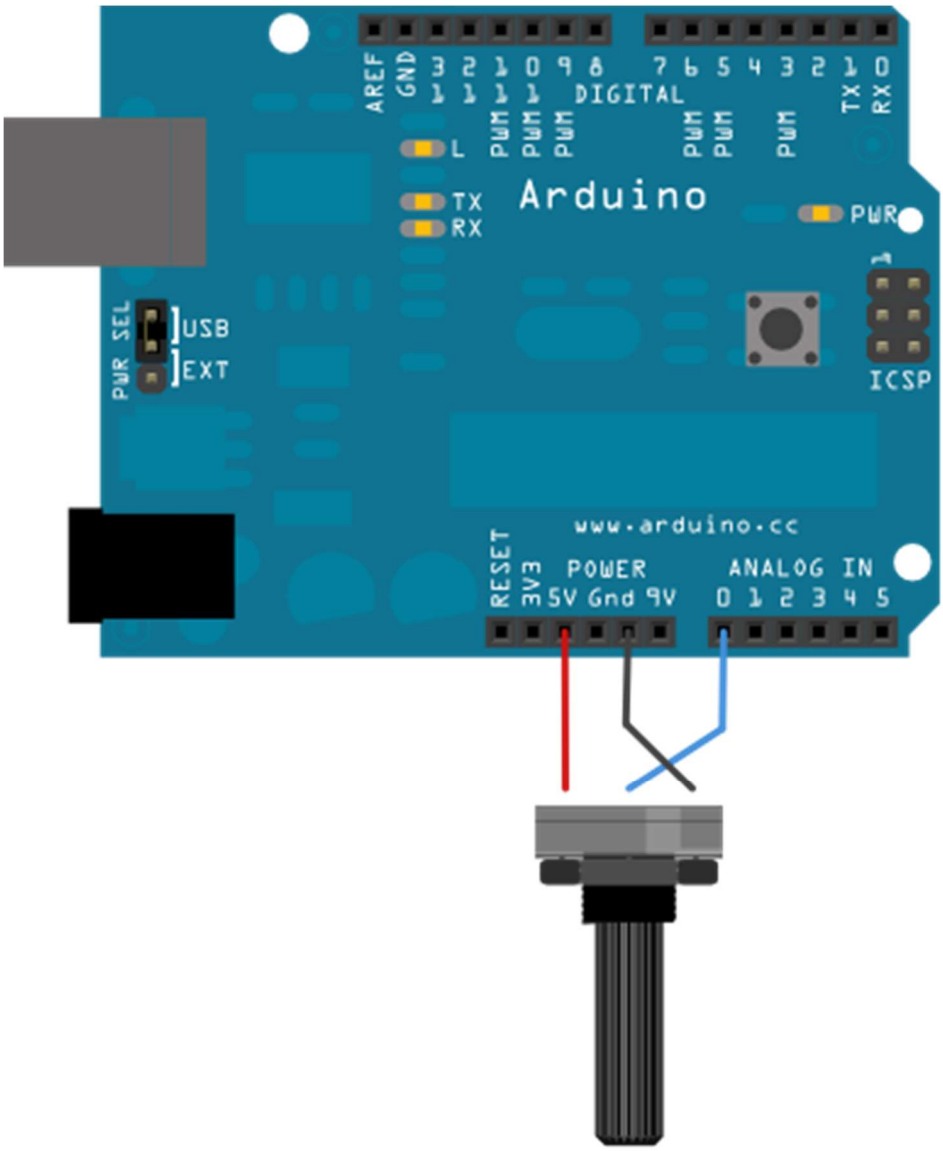
2) Potentiometer *or* 10K ohm photoresistor and 10K ohm resistor

3) built-in LED on pin 13 *or*

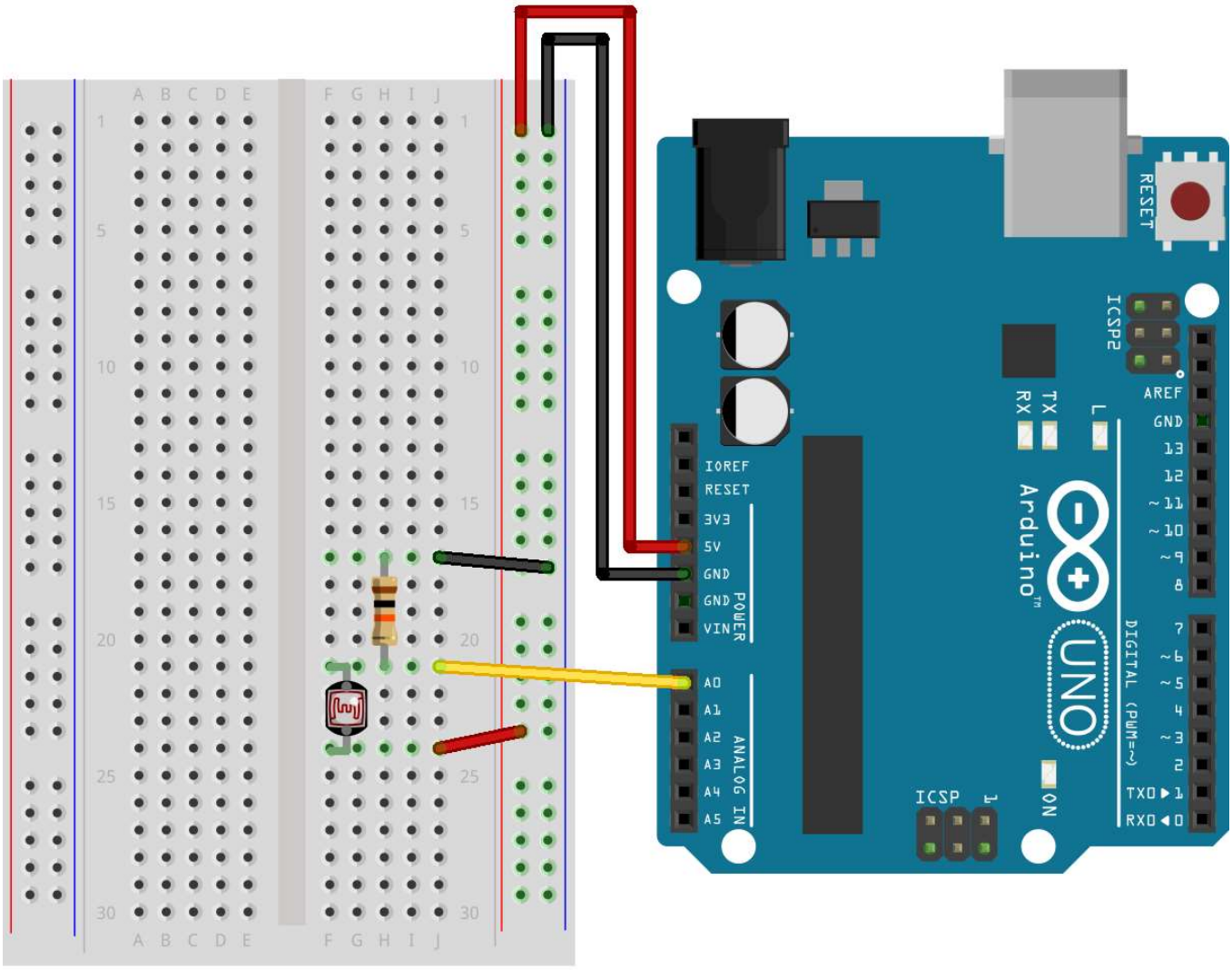
4) 220 ohm resistor and red LED

Circuit

With a potentiometer



With a photoresistor



Connect three wires to the Arduino board. The first goes to ground from one of the outer pins of the potentiometer. The second goes from 5 volts to the other outer pin of the potentiometer. The third goes from analogue input 0 to the middle pin of the potentiometer.

For this example, it is possible to use the board's built in LED attached to pin 13. To use an additional LED, attach its longer leg (the positive leg, or anode), to digital pin 13 in series with the 220 ohm resistor, and it's shorter leg (the negative leg, or cathode) to the ground (GND) pin next to pin 13.

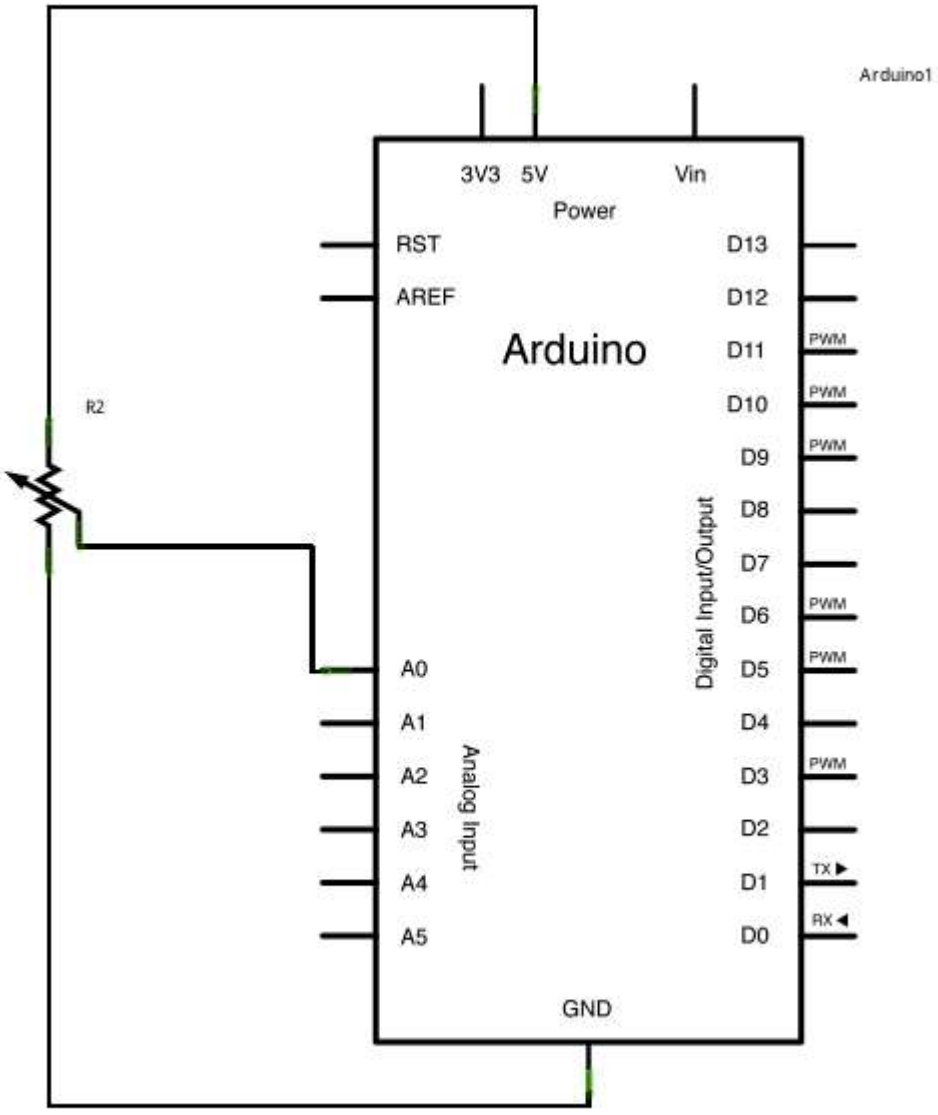
The circuit based on a photoresistor uses a resistor divider to allow the high impedence Analog input to measure the voltage. These inputs do not draw almost any current, therefore by Ohm's law the voltage measured on the other end of a resistor connected to 5V is always 5V, regardless the resistor's value. To get a voltage proportional to the photoresistor value, a resistor divider is necessary. This circuit uses a variable resistor, a fixed resistor and the measurement point is in the middle of the resistors. The voltage measured (Vout) follows this formula:

Vout=Vin*(R2/(R1+R2))

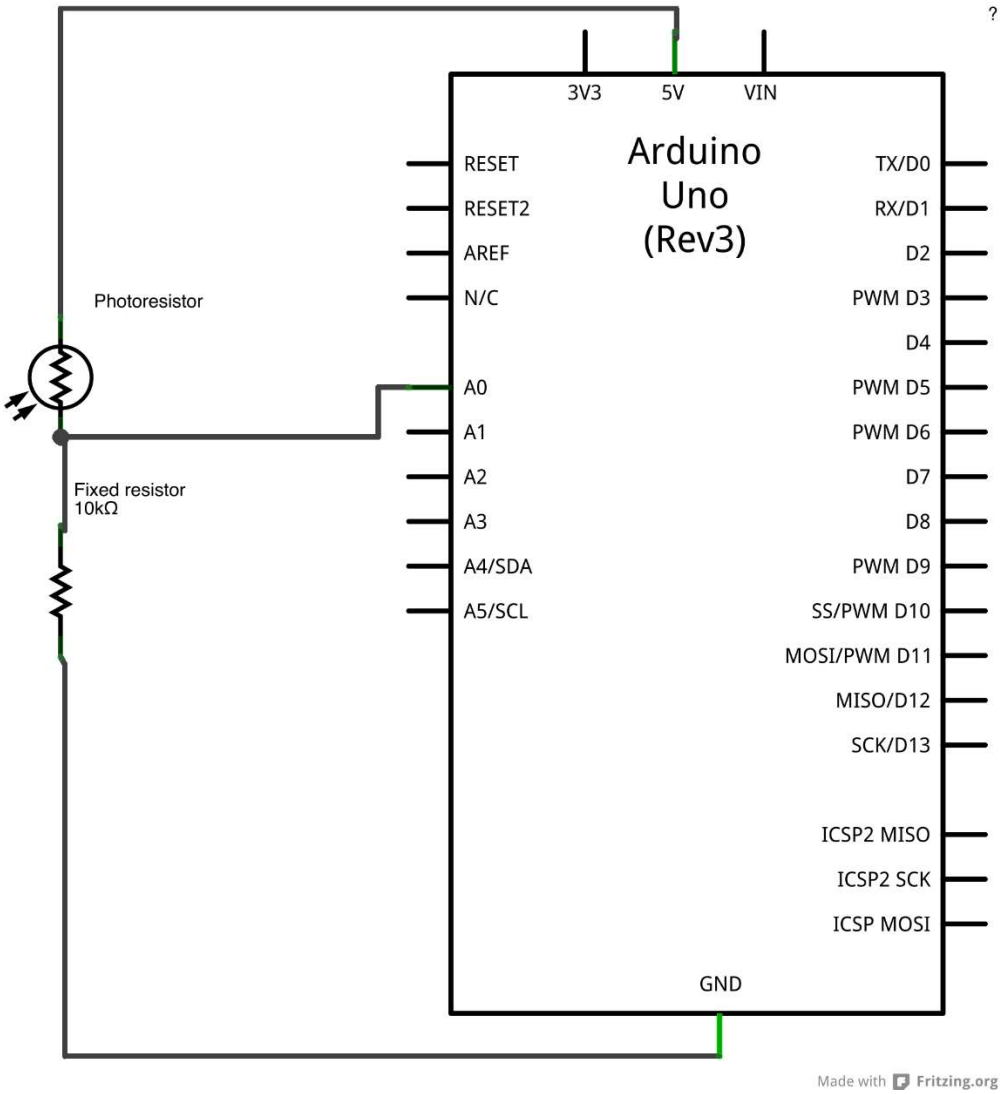
where Vin is 5V, R2 is 10k ohm and R1 is the photoresistor value that ranges from 1M ohm in darkness to 10k ohm in daylight (10 lumen) and less than 1k ohm in bright light or sunlight (>100 lumen).

Schematic

click the image to enlarge
Potentiometer



Photoresistor



Code

At the beginning of this sketch, the variable `sensorPin` is set to analogue pin 0, where your potentiometer is attached, and `ledPin` is set to digital pin 13. You'll also create another variable, `sensorValue` to store the values read from your sensor.

The `analogRead()` command converts the input voltage range, 0 to 5 volts, to a digital value between 0 and 1023. This is done by a circuit inside the microcontroller called an *analogue-to-digital converter* or *ADC*.

By turning the shaft of the potentiometer, you change the amount of resistance on either side of the centre pin (or wiper) of the potentiometer. This changes the relative resistances between the centre pin and the two outside pins, giving you a different voltage at the analogue input. When the shaft is turned all the way in

one direction, there is no resistance between the centre pin and the pin connected to ground. The voltage at the centre pin then is 0 volts, and `analogRead()` returns 0. When the shaft is turned all the way in the other direction, there is no resistance between the centre pin and the pin connected to +5 volts. The voltage at the centre pin then is 5 volts, and `analogRead()` returns 1023. In between, `analogRead()` returns a number between 0 and 1023 that is proportional to the amount of voltage being applied to the pin.

That value, stored in `sensorValue`, is used to set a `delay()` for your blink cycle. The higher the value, the longer the cycle, the smaller the value, the shorter the cycle. The value is read at the beginning of the cycle, therefore the on/off time is always equal.

```
/*
  Analog Input
 Demonstrates analog input by reading an analog sensor on analog pin 0 and
 turning on and off a light emitting diode(LED)  connected to digital pin 13.
 The amount of time the LED will be on and off depends on
 the value obtained by analogRead().

 The circuit:
 * Potentiometer attached to analog input 0
 * center pin of the potentiometer to the analog pin
 * one side pin (either one) to ground
 * the other side pin to +5V
 * LED anode (long leg) attached to digital output 13
 * LED cathode (short leg) attached to ground

 * Note: because most Arduinos have a built-in LED attached
 to pin 13 on the board, the LED is optional.


 */

int sensorPin = A0;    // select the input pin for the potentiometer
int ledPin = 13;      // select the pin for the LED
int sensorValue = 0;  // variable to store the value coming from the sensor

void setup() {
  // declare the ledPin as an OUTPUT:
  pinMode(ledPin, OUTPUT);
}

void loop() {
  // read the value from the sensor:
  sensorValue = analogRead(sensorPin);
  // turn the ledPin on
  digitalWrite(ledPin, HIGH);
  // stop the program for <sensorValue> milliseconds:
  delay(sensorValue);
  // turn the ledPin off:
```

```
  digitalWrite(ledPin, LOW);
  // stop the program for for <sensorValue> milliseconds:
  delay(sensorValue);
}
```