

Potentiometer Analog Input With Arduino (Fatemeh Nabidoust)

Sunday, February 16, 2025 10:59 AM

How To Read a Variable resistor, in this Case a Potentiometer using Arduino's Analog Input.

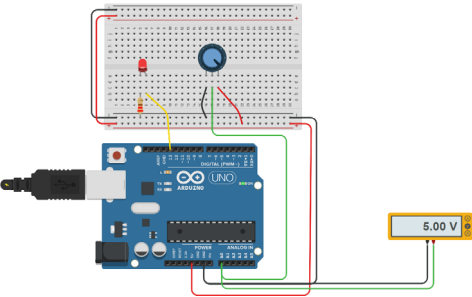
we'll create a circuit and compose a simple program to control the rate of a flashing LED based on the position of a knob we can turn.

we'll build upon the basic circuit for lighting up an LED connected to pin 13.

in the previous lesson, we learned to detect a pushbutton's state, on or off, with digital input. in this lesson, we'll move to the opposite side of the Arduino board to the analog inputs, which sense the gradually changing electrical signal from turning the potentiometer.

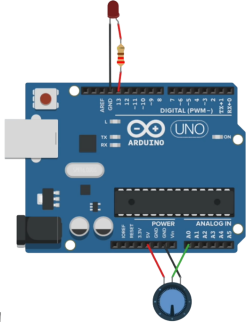


The Arduino's analog to digital converter, or ADC, converts an incoming analog signal between 0V and 5V into a range of numbers from 0-1023.



you can also just click around the potentiometer to move the knob to that position.

Add a potentiometer to the basic LED circuit and wire it up the outer pins to power and ground, and the center pin to Arduino pin A0. since you may just getting the hang of the breadboard, here's the equivalent free-wired version of this circuit for comparison.



Let's Compose some simple Code to read the changing value of the potentiometer by opening up the Arduino IDE.

Create a new variable named sensorValue. After creating our variable sensorValue, inside the setup we use pinMode() function to establish analog pin A0 as an input, and digital pin 13 as an output.

```
// C++ code
//
int sensorValue = 0;
void setup()
{
  pinMode(A0, INPUT);
  pinMode(13, OUTPUT);
}

void loop()
{
  // Read the value from the sensor
  sensorValue = analogRead(A0);
  // Turn the LED on
  digitalWrite(13, HIGH);
  // Pause the program for <sensorValue> milliseconds
  delay(sensorValue); // Wait for sensorValue millisecond(s)
  // Turn the LED off
  digitalWrite(13, LOW);
  // Pause the program for <sensorValue> milliseconds
  delay(sensorValue); // Wait for sensorValue millisecond(s)
}
```

the code inside the loop uses a new function called analogRead() to listen to the pin's state. the program uses a familiar function digitalWrite() and then another one you might know well by now, delay(). but instead of pausing for a fixed amount of time, the value passed to the delay() function will change as you turn the knob because each time through the loop it's going to read the position again. So the LED flashes at a rate of somewhere between 0 and 1023 milliseconds, or just about a second or less.

The basic pushbutton circuit and code is available as a circuit starter. use it anytime you want to read a potentiometer or other kind of variable resistor with analog input.

