Arduino 101
Hands-on: Blink x8

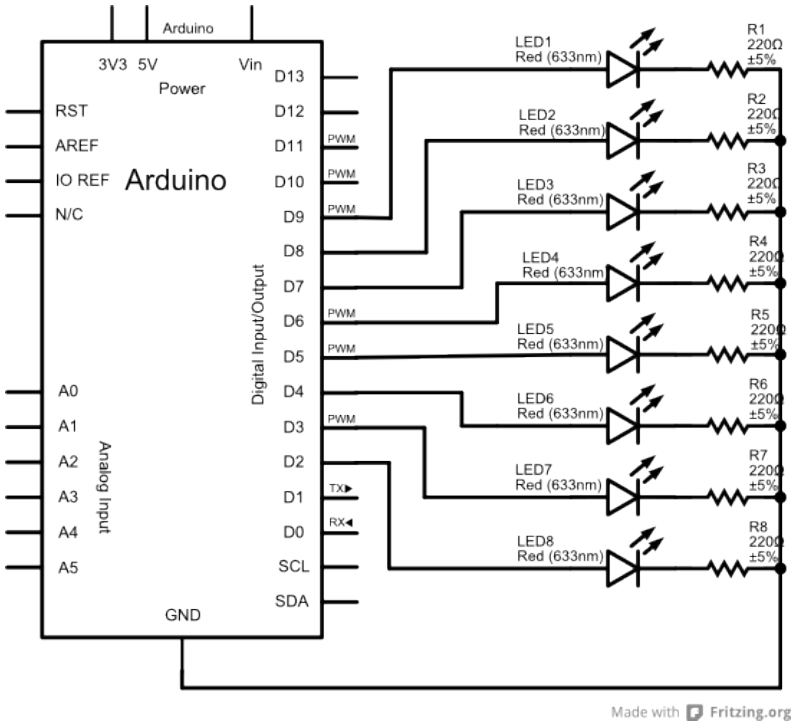
Project Description

This project will expand on the Blink sketch to blink 8 LEDs. It uses the same principle as the original Blink sketch, but it makes use of a **for** loop to initialize the output pin and blink the LEDs in succession.

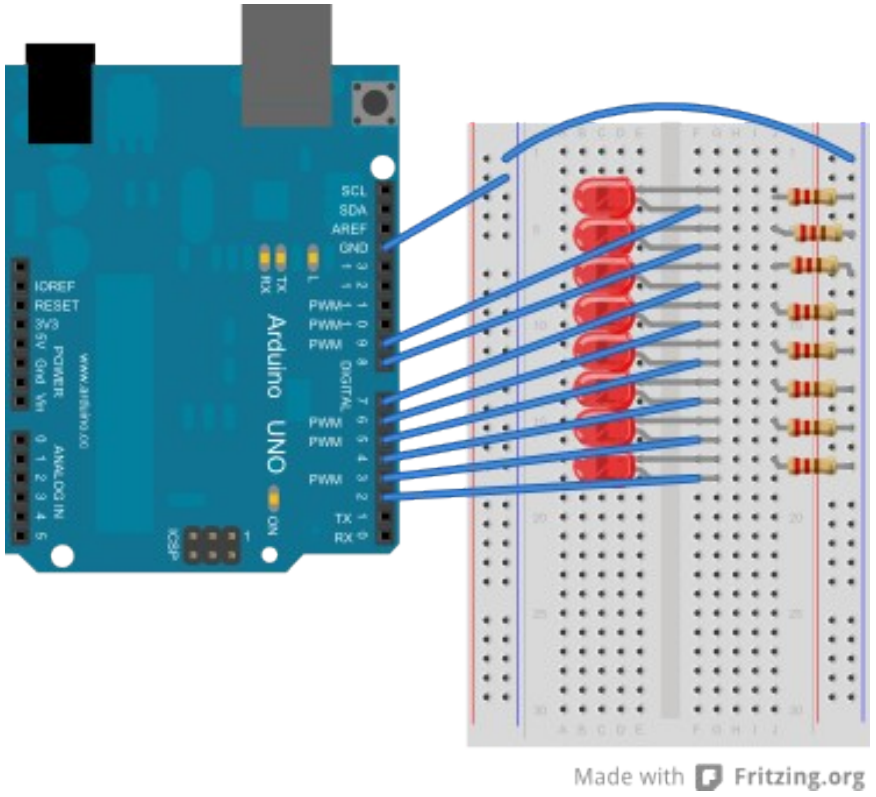
Required Parts

8 red LEDs
8 220Ω Resistors (red, red, black, black)

Schematic



Circuit



NOTES: You are connecting the Arduino pins to the anodes (long lead) of the LEDs. The cathodes (short lead) connect to a 220Ω resistor that connects to ground (blue rail). Don't forget to connect the GND pin from the Arduino to the blue ground rails!

Code

```
// Blink Chaser Sketch for Arduino 101
// by Nick Borko

void setup() {
  int pin;
  // initialize pins 2-9 to be output pins
  for (pin = 2; pin <= 9; pin += 1) {
    pinMode(pin, OUTPUT);
  }
}

void loop() {
  int pin;
  // loop from 2 to 8
  for (pin = 2; pin < 9; pin += 1) {
    digitalWrite(pin, HIGH);
    delay(100);
    digitalWrite(pin, LOW);
  }
  // loop again from 9 to 3
  for (pin = 9; pin > 2; pin -= 1) {
    digitalWrite(pin, HIGH);
    delay(100);
    digitalWrite(pin, LOW);
  }
}
```

Discussion

The theory behind the circuit and code is the same as the original Blink sketch. The main difference is that instead of writing the same code 8 times (once for each LED), we use a **for** loop with a variable that contains the number of the **pin** to work on.

For example, in the **setup()** function, the for loop sets a variable **pin** to a value of 2, and continues the loop until its value is 10, at which point it fails the test of **pin <= 9**. The loop statement increments the value of **pin** by one after each execution of the loop body, which sets the **pinMode** for the **pin** to **OUTPUT**.

The same sort of thing happens in the **loop()** function, except that the loop begins and ends at different values to give the "chaser" effect. See if you can follow the logic of each **for** loop to understand how it works.

You might ask, why is the **pin** variable declared in both the **setup()** and **loop()** functions? The reason is that each function has its own **scope**. That means that variables declared in a function are seen only by the code in that function and are said to be are **local** to the function. If **pin** had been declared outside either of these functions, then it would become a **global** variable and could be used in both functions.