Digital output

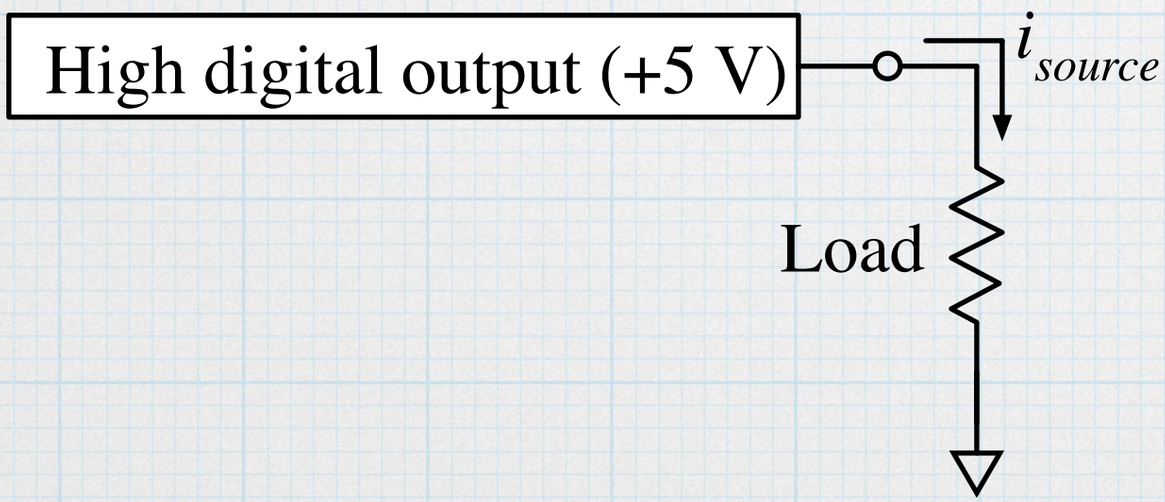
- The digital pins on the Arduino can be set to either read a digital input or write a digital output. We will discuss the two modes in separate documents.
- A digital output can be used to control LEDs, switches (transistors or relays), displays, and for communication channels.
- The digital output levels are: low ≈ 0 V and high $\approx$ power supply voltage (V_{CC}, either 5 V or 3.3 V).
- Two steps are requires: first set the input/output mode to output with `pinMode( pin, OUTPUT )`, where `pin` is the number of the specific digital pin (`OUTPUT` is a standard keyword). This command is usually in the setup function, but can be in the repeat function if needed.
- Then set the level: `digitalWrite( pin, value )`, where `pin` is the pin number and `value` is either `HIGH` or `LOW` (standard keywords). Usually this is in the loop function, but can be in either.

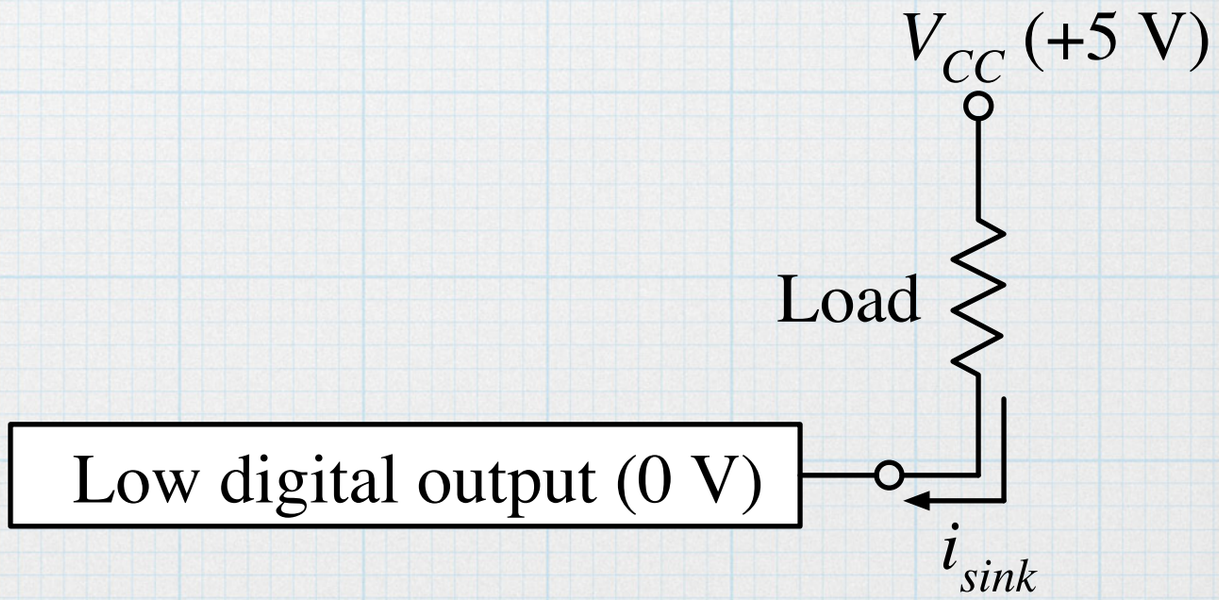
Example

```
//Generate a simple 500 Hz square wave. (T = 2 ms)

int swPin = 3;   //Pin 3 will be used for the square wave

void setup()
{
  pinmode( swPin, OUTPUT );   //define pin 3 to be an output
}

void repeat()
{
  digitalWrite( swPin, HIGH);   //Make it high.
  delay( 1 );                   //Wait 1 ms
  digitalWrite( swPin, LOW);    //Make it low.
  delay( 1 );                   //Wait 1 ms
}
```

Current limits

- The digital pins can source or sink a maximum of 40 mA. Probably close to the point of causing damage to internal circuitry.
- As a practical rule, probably should limit currents to 20 mA for safety.
- If more current is needed to control a load, use a transistor or relay.
- Note: Max current allowed through each supply pin is 200 mA. There is one V_{CC} pin so 200 mA max to source for the whole chip. Two ground pins, so 400 mA max to sink for the chip.
- Ref: http://playground.arduino.cc/Main/ArduinoPinCurrentLimitations

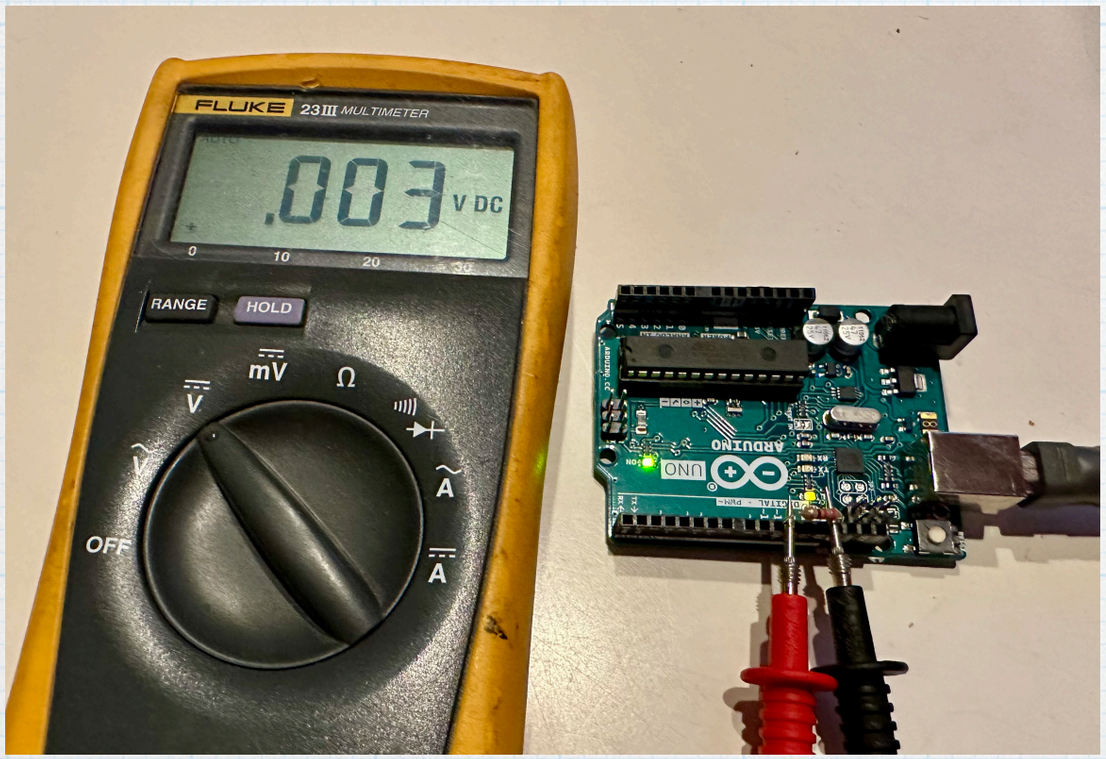




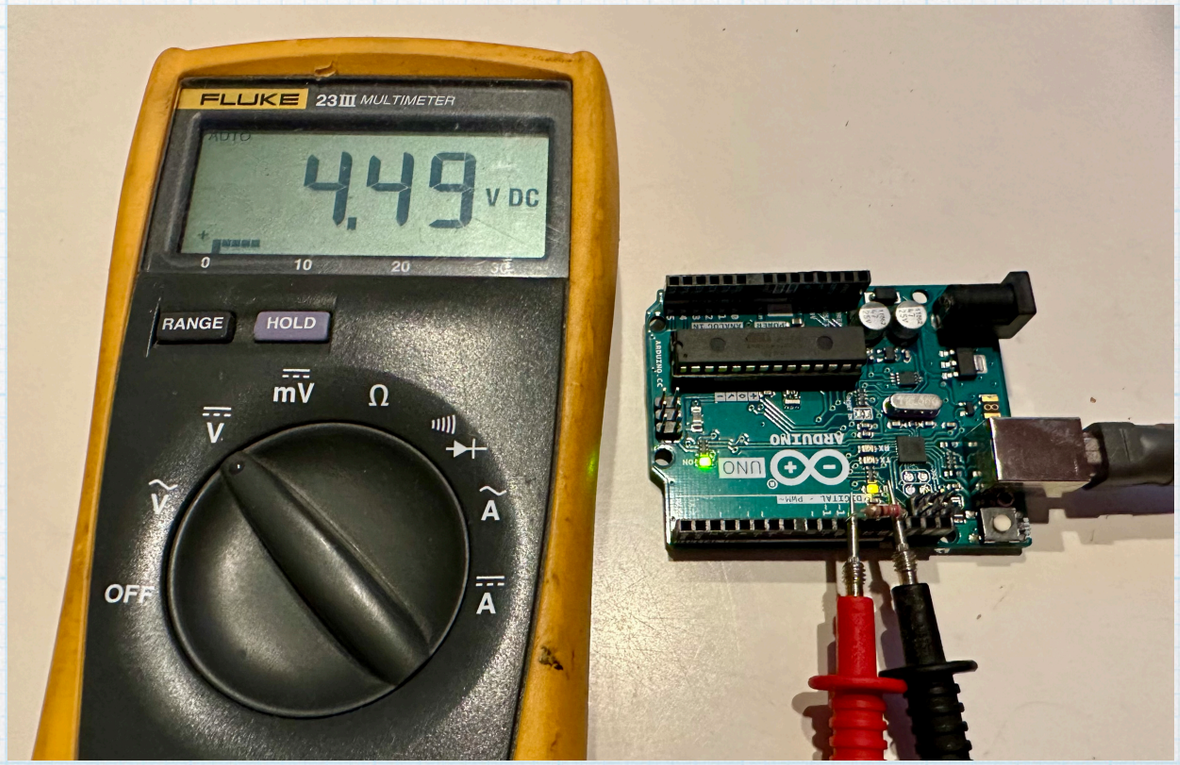
Check

Attach a 220-Ω resistor between digital pin 10 and ground. V_{CC} = 5 V.

Then set pin 10 to `LOW`. Pretty close to zero.



Then set it `HIGH`. Should be about 22 mA of current. Not equal to V_{CC}, but still "high" as a digital signal.

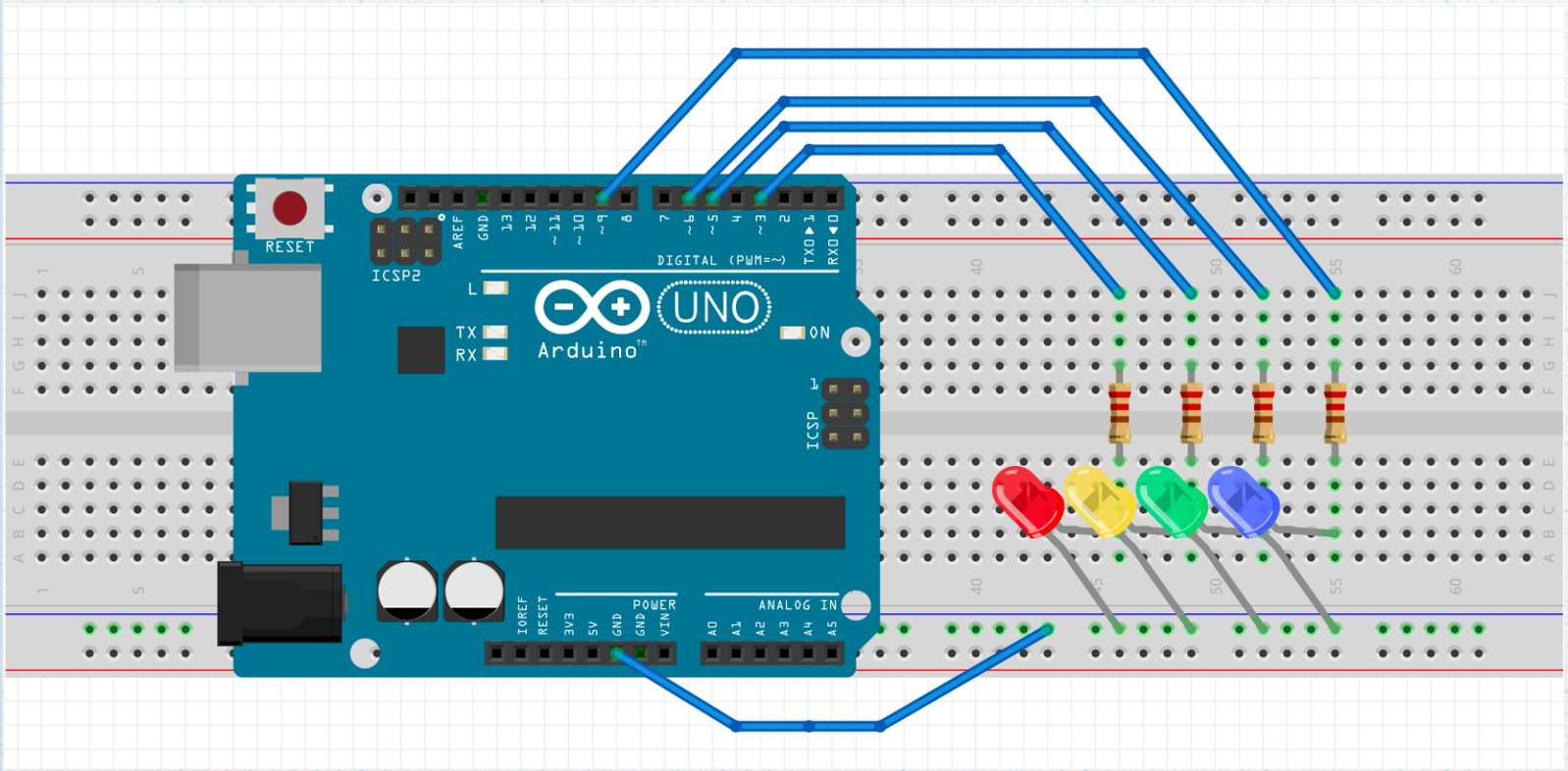


Usually, the more current being sourced, the lower the output will be.

Example: Blink some LEDs

A very simple way to demonstrate digital outputs is using LEDs. We will use 4 LEDs and blink them sequentially.

Use digital pins 3, 5, 6, and 9. Don't forget the current-limit resistors! (Each is 220 Ω here.)



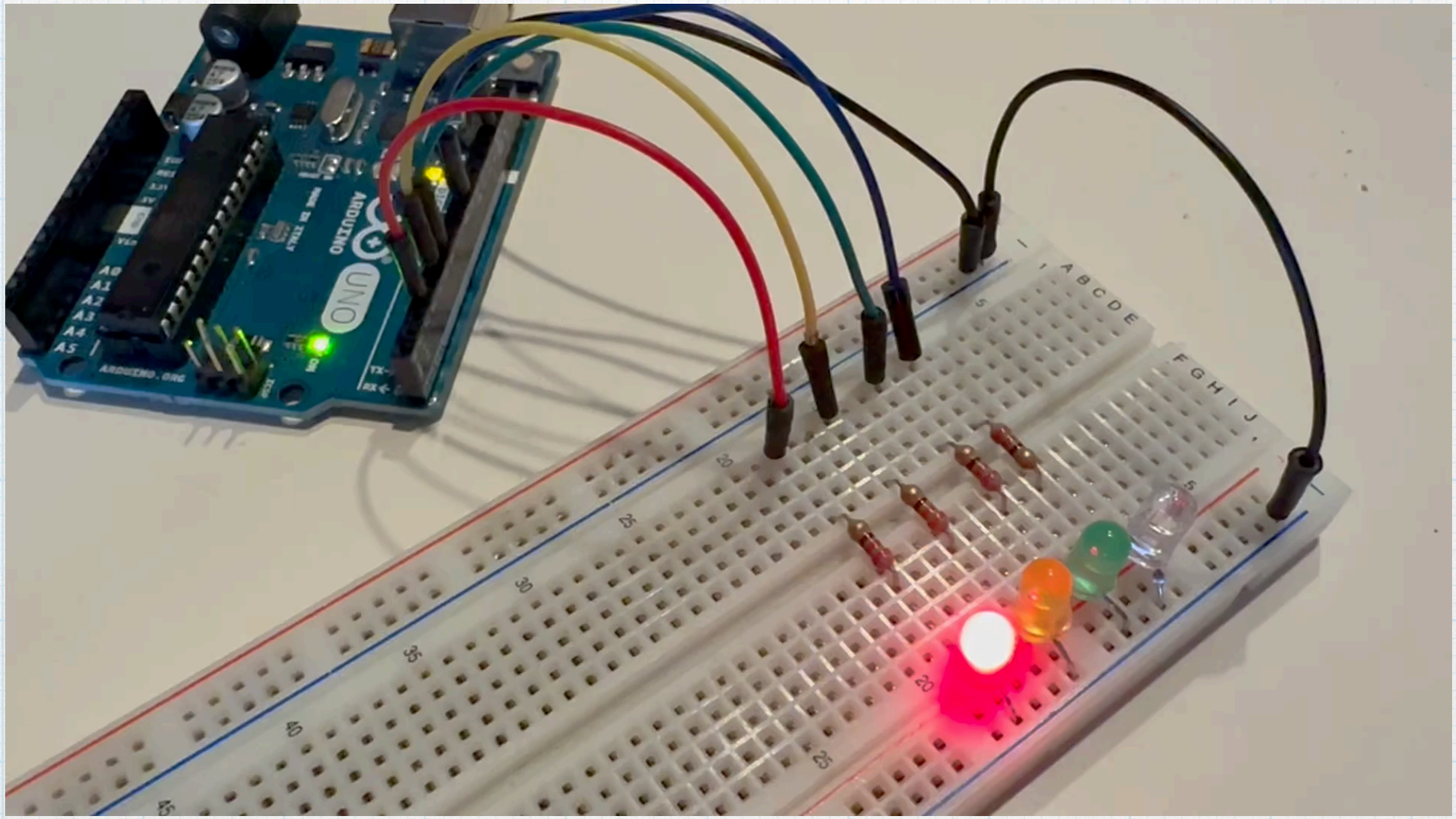
To make these diagrams:
https://fritzing.org .
$10 donation.

Code

four_leds.ino

```
1   // Fun with 4 LEDs
2   int i;
3   int pin[4];             //digital out pins
4   int loopTime = 500;     //delay time between loops, in millisec
5   int onTime = 500;       //diode on time, in millisec
6   int offTime = 500;      //diode off time, in millisec
7
8   void setup()
9   {
10    //Set up digital pins 3, 5, 6, and 9 for output.
11    pin[0] = 3;
12    pin[1] = 5;
13    pin[2] = 6;
14    pin[3] = 9;
15
16    for( i = 0; i <=3; i++ )
17      pinMode( pin[i], OUTPUT );
18  }
19
20  void loop()
21  {
22    for( i = 0; i <=3; i++ )
23    {
24      digitalWrite( pin[i], HIGH );
25      delay( onTime );
26      digitalWrite( pin[i], LOW );
27      delay( offTime );
28    }
29
30    delay( loopTime );
31  }
```

Not very exciting, but it works as advertised. Most importantly, it is quite easy to control things with digital outputs. Just be mindful of the currents.



It is easy to change the flashing pattern by adjusting the on, off, and loop times.