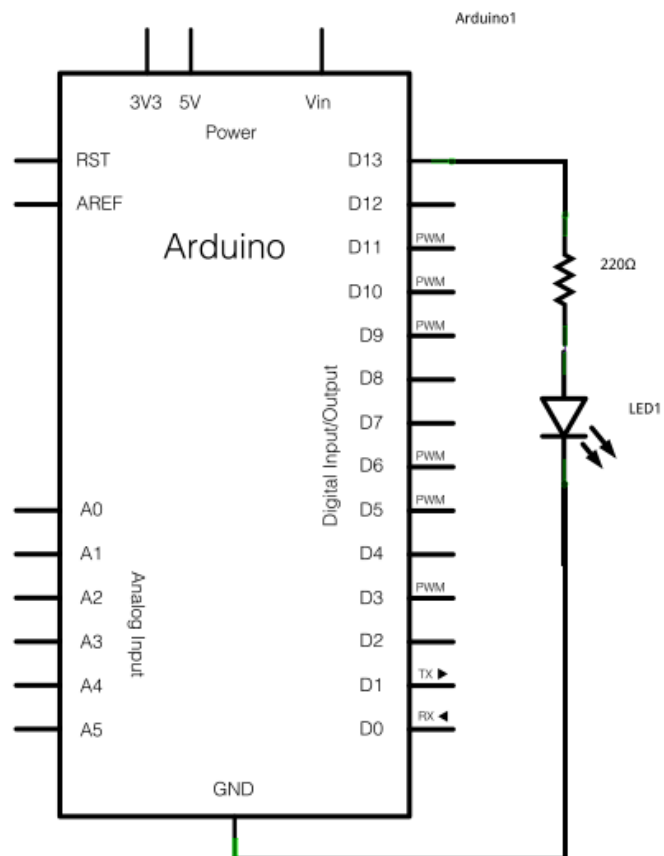


Blink Without Delay

Schematic



Code

The code below uses the `millis()` function, a command that returns the number of milliseconds since the board started running its current sketch, to blink an LED.

```
// constants won't change. Used here to set a pin number:
const int ledPin = 13;    // the number of LED pin

int ledState = LOW;

unsigned long previousMillis = 0; // will store last time LED was updated

// constants won't change:
const long interval = 1000;    // interval at which to blink (milliseconds)

void setup() {
  // set the digital pin as output:
  pinMode(ledPin, OUTPUT);
}

void loop() {

  unsigned long currentMillis = millis();

  if (currentMillis - previousMillis >= interval) {
    // save the last time you blinked the LED
    previousMillis = currentMillis;

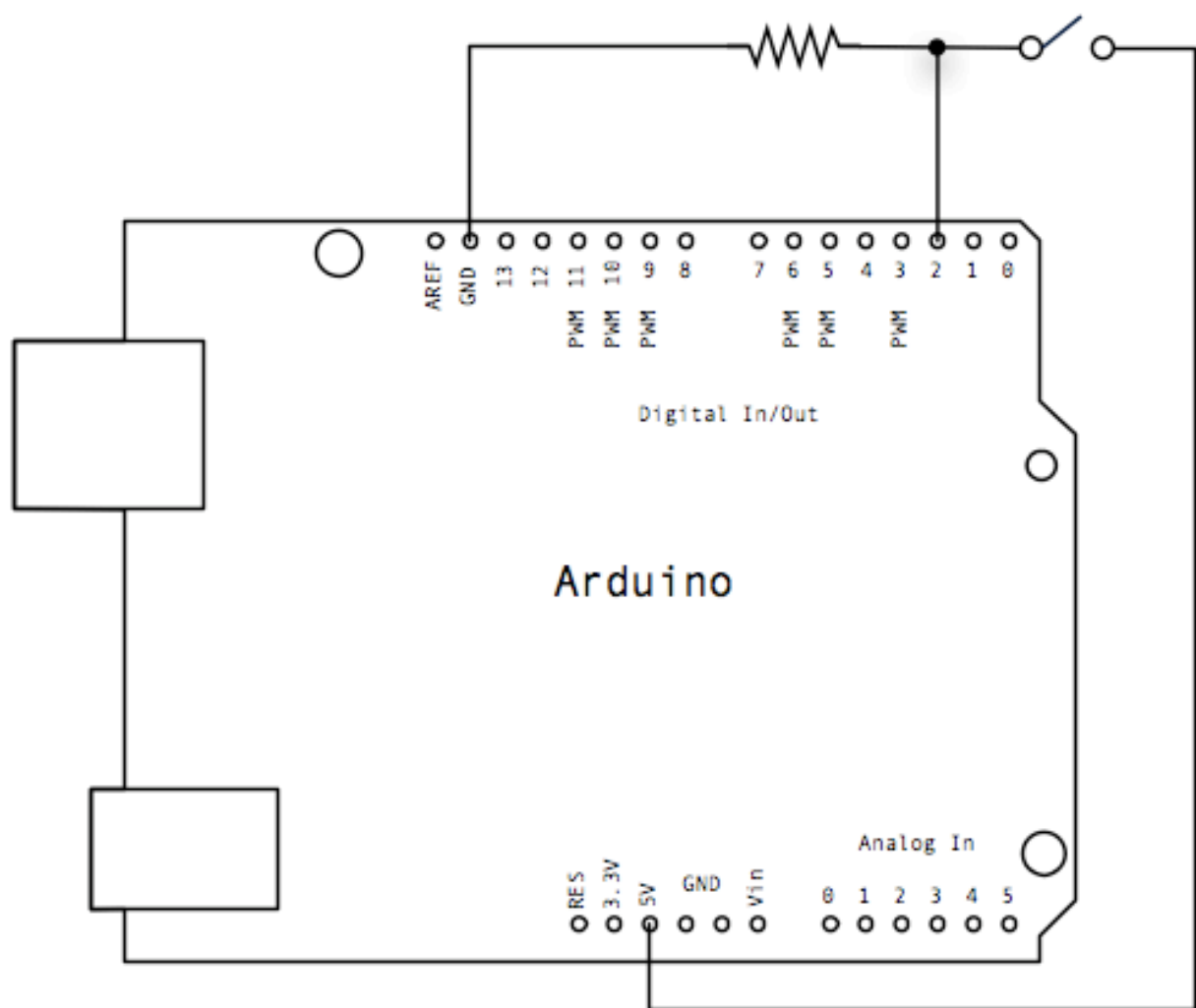
    // if the LED is off turn it on and vice-versa:
    if (ledState == LOW) {
      ledState = HIGH;
    } else {
      ledState = LOW;
    }

    // set the LED with the ledState of the variable:
    digitalWrite(ledPin, ledState);
  }

}
```

Digital read

Schematic



Code

The code below uses the `Serial.print()` function, a command that prints data to the serial port as human-readable ASCII text. This command can take many forms. Numbers are printed using an ASCII character for each digit. Floats are similarly printed as ASCII digits, defaulting to two decimal places. Bytes are sent as a single character. Characters and strings are sent as is. For example:

- `Serial.print(78)` gives "78"
- `Serial.print(1.23456)` gives "1.23"
- `Serial.print('N')` gives "N"
- `Serial.print("Hello world.")` gives "Hello world."

```
// constants won't change. They're used here to set pin numbers:
const int buttonPin = 2;    // the number of the pushbutton pin

void setup() {

  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);

  Serial.begin(9600);    // open the serial port at 9600 bps:
}

void loop() {

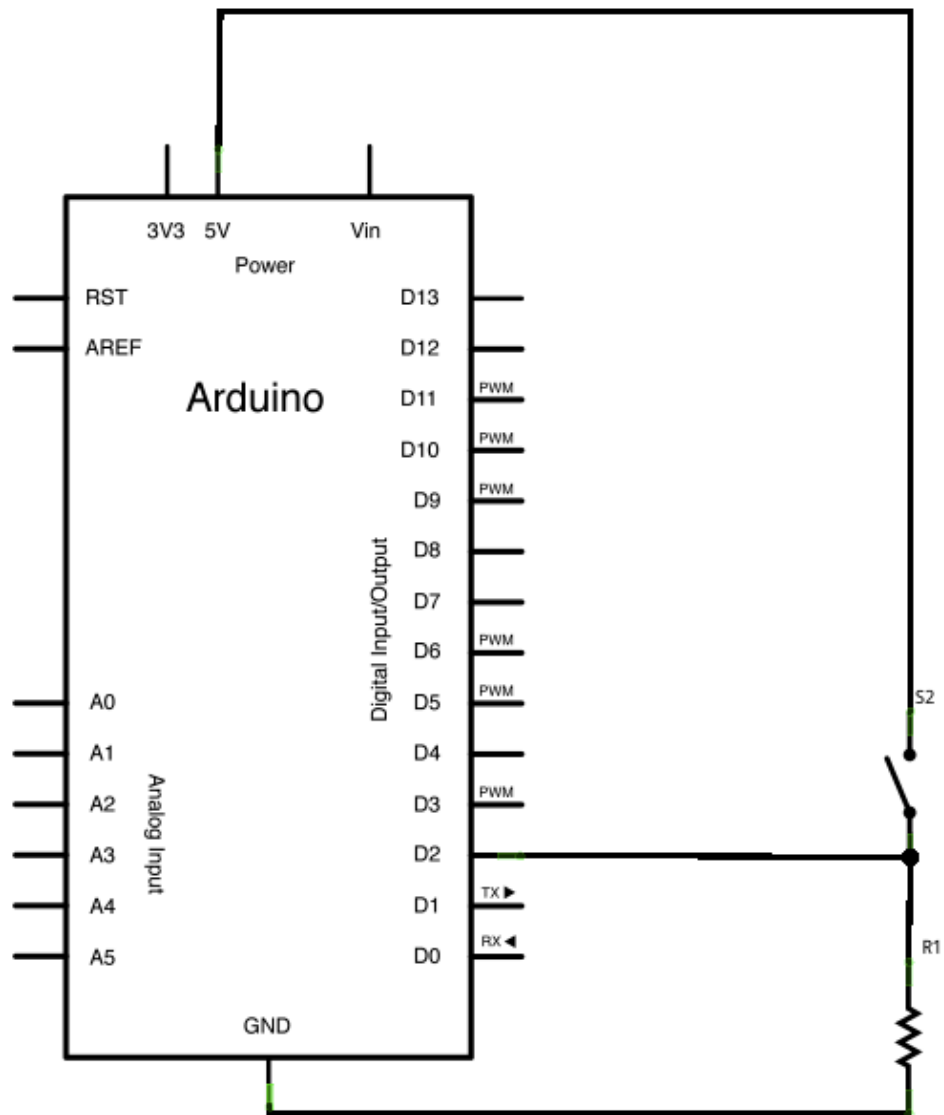
  // read the state of the pushbutton value:
  int buttonState = digitalRead(buttonPin);

  Serial.println(buttonState);

}
```

Debounce

Schematic



Controls LED with button

Code

```
// constants won't change. They're used here to set pin numbers:
const int buttonPin = 2;    // the number of the pushbutton pin
const int ledPin = 13;      // the number of the LED pin

void setup() {

  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);

  // set the digital pin as output:
  pinMode(ledPin, OUTPUT);

}

void loop() {

  // read the state of the pushbutton value:
  int buttonState = digitalRead(buttonPin);

  digitalWrite(ledPin, buttonState);

}
```