Blink Without Delay

Sometimes you need to do two things at once. For example, you might want to blink an LED while reading a button press. In this case, you can't use `delay()`, because Arduino pauses your program during the `delay()`. If the button is pressed while Arduino is paused waiting for the `delay()` to pass, your program will miss the button press.

This sketch demonstrates how to blink an LED without using `delay()`. It turns on the LED on and then makes note of the time. Then, each time through `loop()`, it checks to see if the desired blink time has passed. If it has, it toggles the LED on or off and makes note of the new time. In this way the LED blinks continuously while the sketch execution never lags on a single instruction.

An analogy would be warming up a pizza in your microwave, and also waiting some important email. You put the pizza in the microwave and set it for 10 minutes. The analogy to using `delay()` would be to sit in front of the microwave watching the timer count down from 10 minutes until the timer reaches zero. If the important email arrives during this time you will miss it.

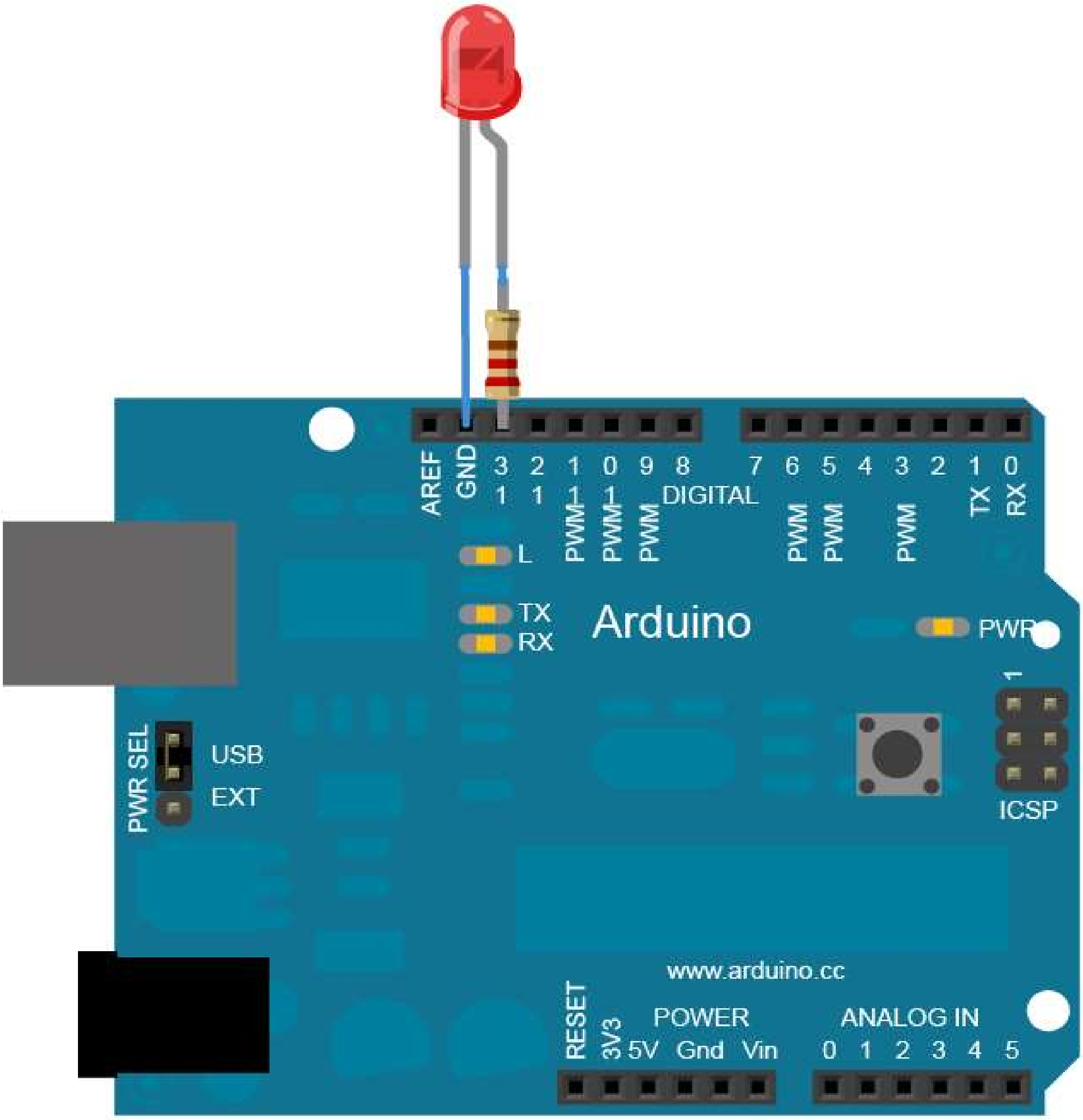
What you would do in real life would be to turn on the pizza, and then check your email, and then maybe do something else (that doesn't take too long!) and every so often you will come back to the microwave to see if the timer has reached zero, indicating that your pizza is done.

In this tutorial you will learn how to set up a similar timer.

Hardware Required

1) Arduino Board
2) LED
3) 220 ohm resistor

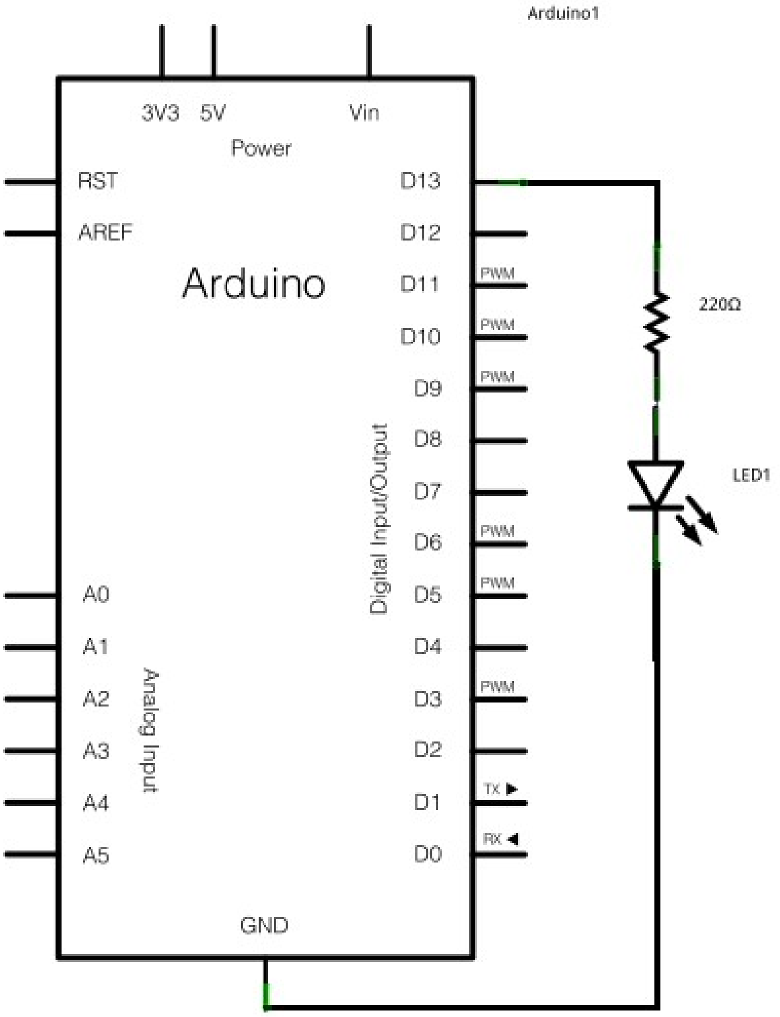
Circuit



To build the circuit, connect one end of the resistor to pin 13 of the board. Connect the long leg of the LED (the positive leg, called the anode) to the other end of the resistor. Connect the short leg of the LED (the negative leg, called the cathode) to the board GND, as shown in the diagram above and the schematic below.

Most Arduino boards already have an LED attached to pin 13 on the board itself. If you run this example with no hardware attached, you should see that LED blink.

Schematic



After you build the circuit plug your board into your computer, start the Arduino Software (IDE), and enter the code below.

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.

```
/* Blink without Delay
 Turns on and off a light emitting diode (LED) connected to a digital
 pin, without using the delay() function.  This means that other code
 can run at the same time without being interrupted by the LED code.
 The circuit:
 * LED attached from pin 13 to ground.
 * Note: on most Arduinos, there is already an LED on the board
 that's attached to pin 13, so no hardware is needed for this example.
 */

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

// Variables will change :
int ledState = LOW;             // ledState used to set the LED

// Generally, you should use "unsigned long" for variables that hold time
// The value will quickly become too large for an int to store
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() {
  // here is where you'd put code that needs to be running all the time.

  // check to see if it's time to blink the LED; that is, if the
  // difference between the current time and last time you blinked
  // the LED is bigger than the interval at which you want to
  // blink the LED.
  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);
  }
}
```