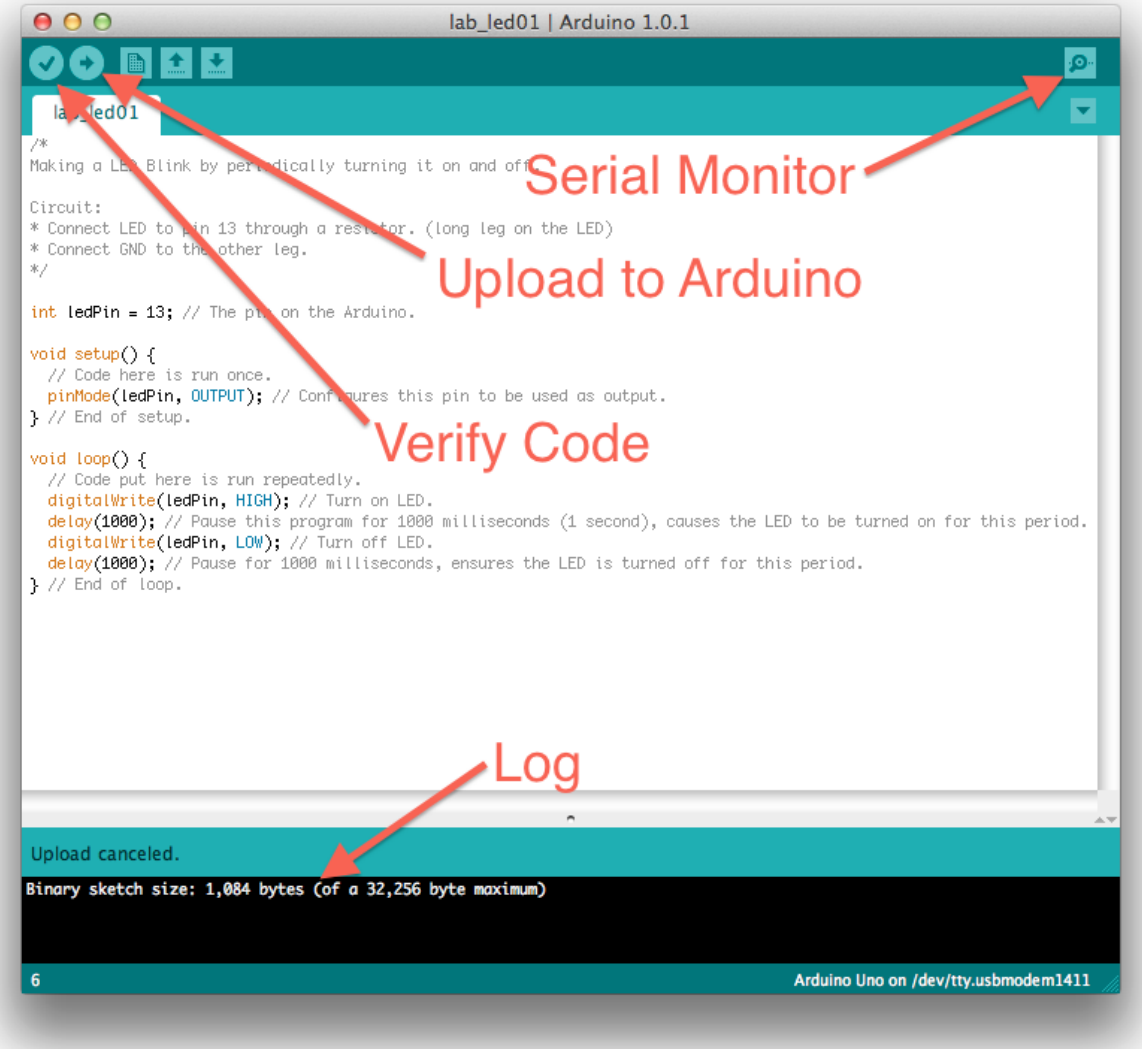
Arduino

Arduino Development Environment

A more comprehensive introduction to the Arduino Enviroment can be found at http://arduino.cc/en/Guide/Environment.

Below is an image that highlights the essential parts.

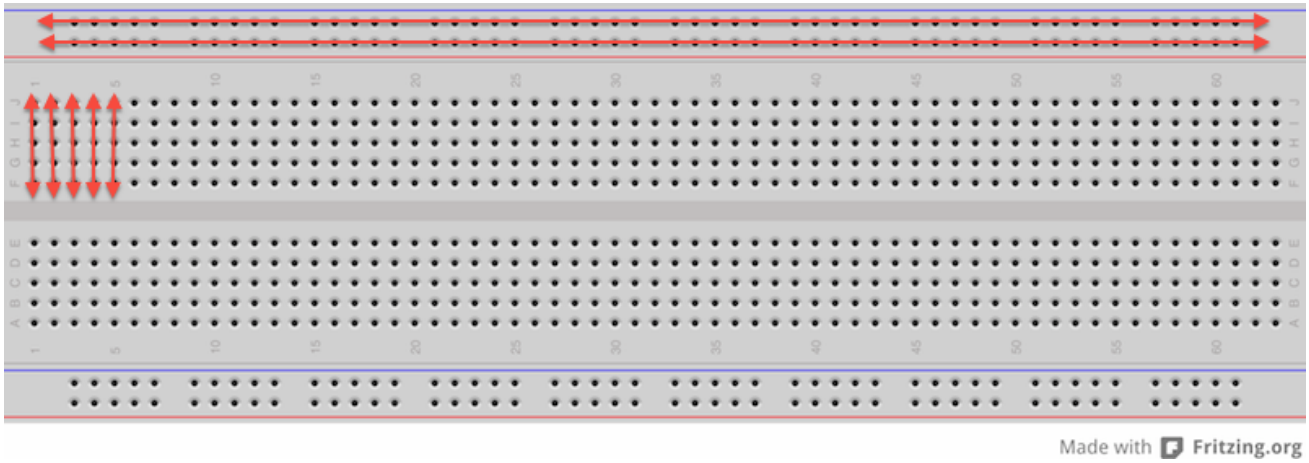


- Verify code is used to check that the code can be run on the Arduinio board before uploading it.
- Upload to Arduino, verifies and uploads it to the Arduino board.
- Serial monitor is used to see output from a program.
- Log shows when code has been uploaded, any errors during a verification shows up here too.

Breadboard

A breadboard is used when wiring components together. This allows us to prototype a circuit without soldering components together.

The top part of the board connects the pins together horizonatally. The two middle sections of the board connects the pin vertically. This is illustrated using the arrows on the board. The top section



When wirering components, make sure power is turned off, e.g. usb cable not plugged into the arduino.

Examples

These examples shows various setups using input and output using the provided components.

Each example provides a code snippet and an image explaining how to wire the components together. Each code snippet also includes a few comments that explains what happens.

Input

Photocell (LDR) - Sensor that that messaures the current light condition.
Flex Resistor - Sensor that reports a value when the sensor is being bend.
Potentiometer - A component that reports a value within a range when it is turned.
Button - A button
PIR - A motion detection sensor, can detect if a dog walks by the sensor.

Output

LED - A diode, usually used as an small indicator lamp.
Piezo - A component that can make noise.
Servo - A rotating motor, great when building robots.

A set of resistors to protect components and to stabilize sensor values.

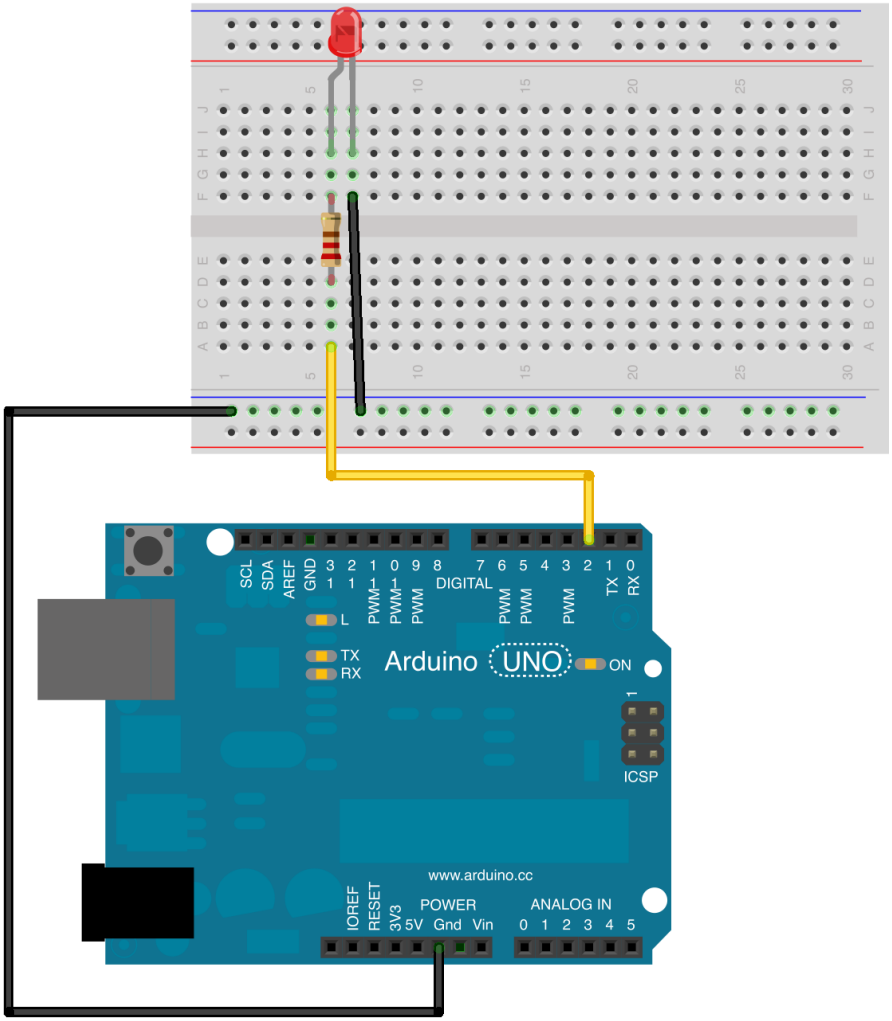
Resistors are color coded, to help map them to a value the following chart is helpful.
http://learn.adafruit.com/multimeters/resistance

LED Blink.

Making a LED blink.

Components

- LED
- Resistor



Add the following code to the Arduino Development Environment and upload it to the Arduino board.

```
int ledPin = 2; // The pin on the Arduino.

void setup() {
```

```
  // Code here is run once.
  pinMode(ledPin, OUTPUT); // Configures this pin to be used as output.
} // End of setup.

void loop() {
  // Code put here is run repeatedly.
  digitalWrite(ledPin, HIGH); // Turn on LED.
  delay(1000); // Pause this program for 1000 milliseconds (1 second), causes the
LED to be turned on for this period.
  digitalWrite(ledPin, LOW); // Turn off LED.
  delay(1000); // Pause for 1000 milliseconds, ensures the LED is turned off for
this period.
} // End of loop.
```

https://github.com/expact/arduinolab/blob/master/lab_led01/lab_led01.ino.

If the LED does not blink, verify wires and check that the long leg of the LED is connected to pin 13.

Input value

This demo shows how we can read values from a sensor and see the result through the Arduino tool Serial Monitor.

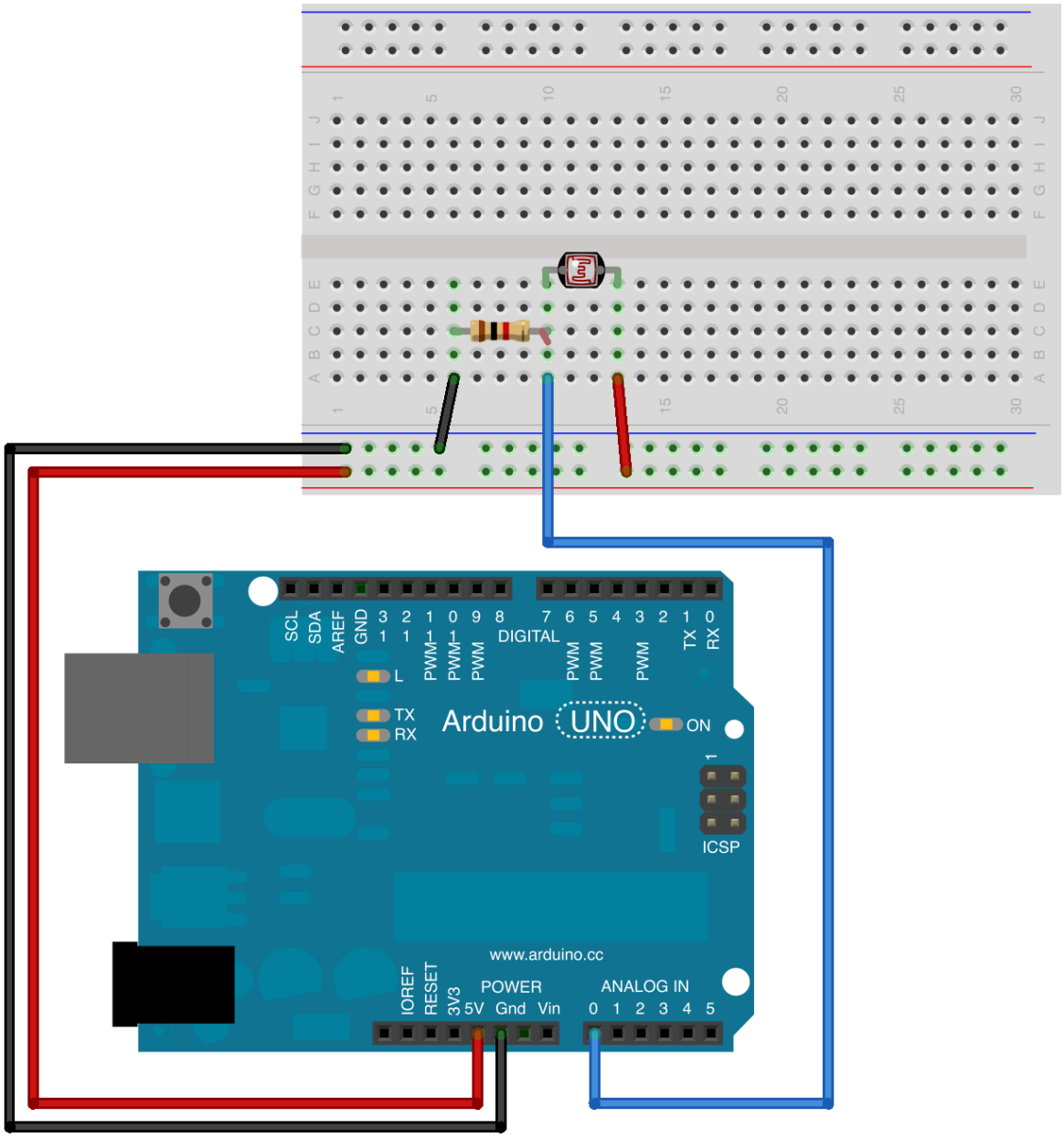
Components

- LDR or Flex Resistor
- Resistor 1k

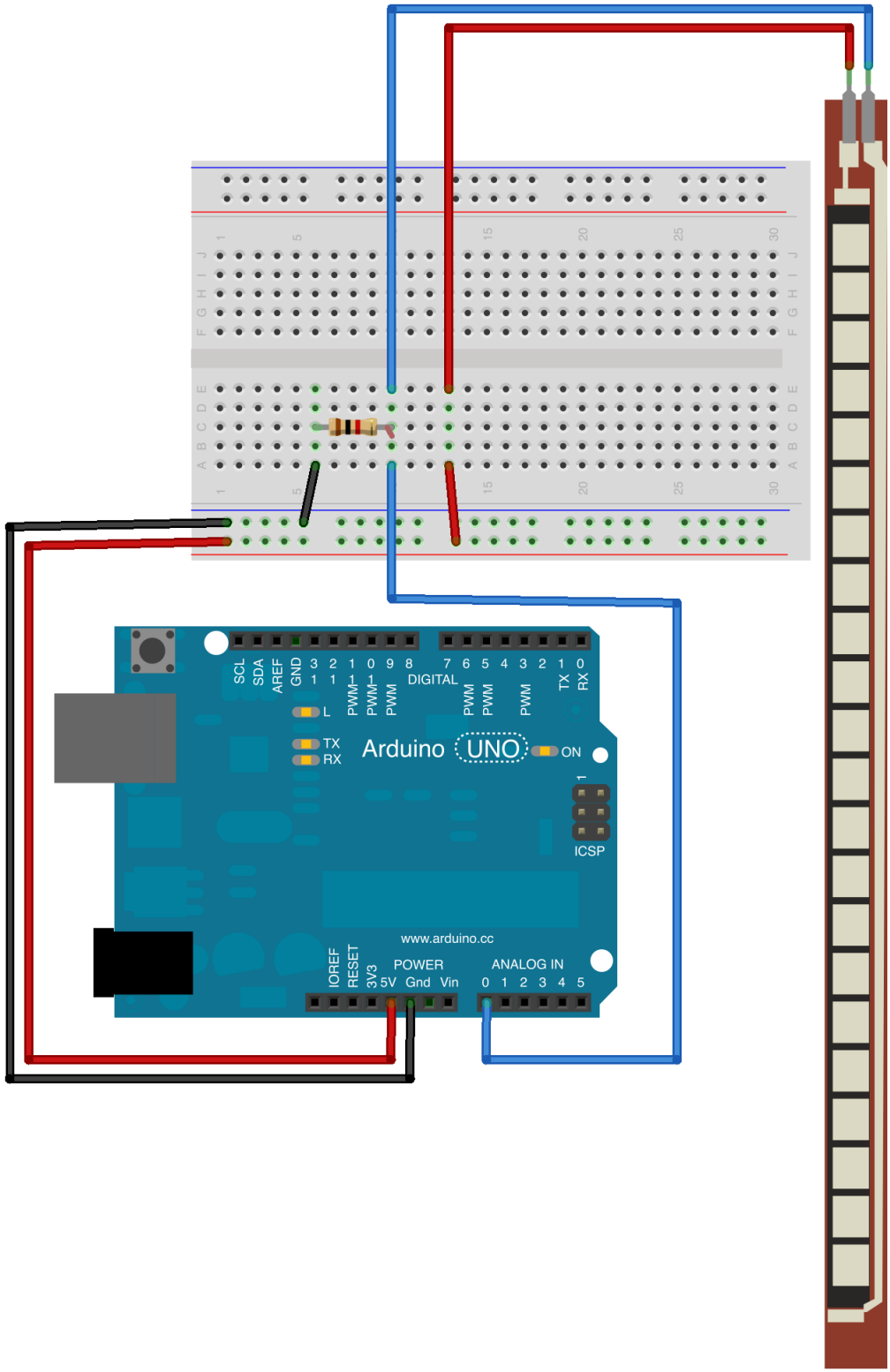
```
int sensorPin = A0; // The analog pin on the Arduino that the sensor is connected
to.
int sensorData = 0; // Variable of where the sensor data is collected.

void setup() {
  Serial.begin(9600); // Allows us to use the Serial Monitor to watch the values.
}

void loop() {
  sensorData = analogRead(sensorPin); // Read the data from tne sensor.
  Serial.println(sensorData); // Print the data to the Serial Monitor so that we
can watch it.
  delay(10); // Pause to make it easier to read the values.
}
```

https://github.com/expact/arduinolab/blob/master/lab_sensor01/lab_sensor01.ino

This example is using a Photo-Resistor (LDR) that can be used in circuits to detect various light conditions.



This example shows how to wire the Flex Resistor.

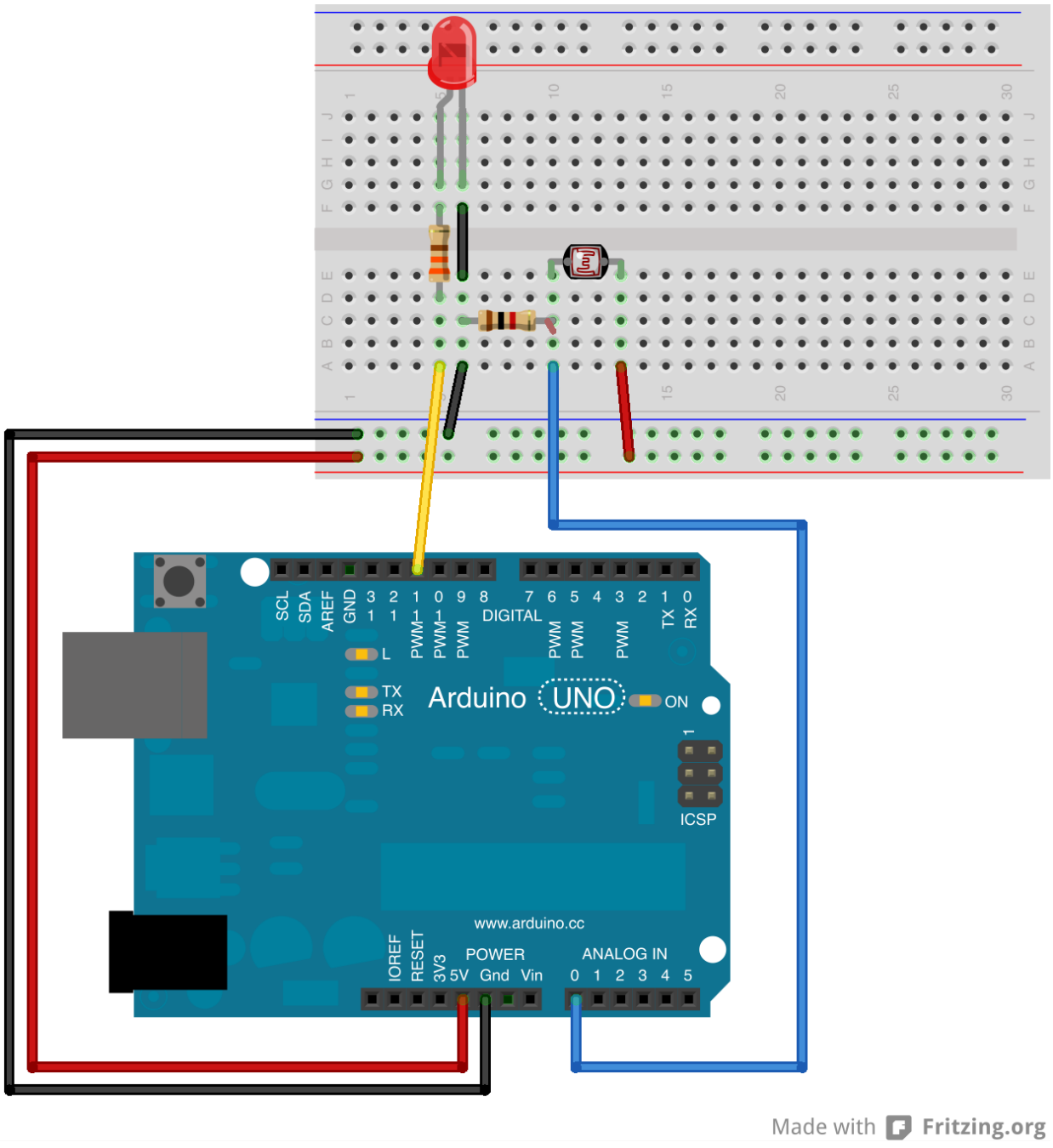


Use the input value with output

This demo builds upon the previous two demos and shows how we can use the input value to to do something useful, like blinking a LED.

Components

- LED
- LDR
- Resistor 1k
- Resistor 220



Note: It's possible to use another resistor for the sensor but the code might need to be adjusted. Use the Serial Monitor to see what the sensor reports.

The follwing code example shows how we can use the sensor value and trigger the led when a certain value is set. Use the Serial Monitor to check what values is being reported from the sensor and adjust the code with different thresholds to see how it links together.

```
int sensorPin = A0; // The analog pin that the sensor is connected to.
int ledPin = 11; // The pin where the LED is connected.
int sensorData = 0; // Variable of where the sensor data is collected.

void setup() {
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600); // Allows us to use the Serial Monitor to watch the values.
}

void loop() {
  sensorData = analogRead(sensorPin); // Read the data from tne sensor.
  Serial.println(sensorData); // Print the data to the Serial Monitor so that we
can watch it.

  if (sensorData < 600) { // If the sensor value is lower than 600, turn on the
LED.
    digitalWrite(ledPin, HIGH);
  } else { // If the sensor value is above 600 turn off the LED.
    digitalWrite(ledPin, LOW);
  }
}
```

https://github.com/expact/arduinolab/blob/master/lab_sensor02/lab_sensor02.ino

The following code examples shows how we can feed the sensor value to a led to adjust the brightness based on the input from the LDR.

```
int sensorPin = A0; // The analog pin that the sensor is connected to.
int ledPin = 11; // The pin where the led is connected, needs to be one of the
PWMs.
int sensorData = 0; // Variable of where the sensor data is collected.
int brightness = 0; // Brightness of the LED, 255 is full brigthness and 0 is
completly off.

void setup() {
  pinMode(ledPin, OUTPUT);
}

void loop() {
  sensorData = analogRead(sensorPin); // Read the data from tne sensor.
```

```
  // Re-map the sensor value from the range 0, 1023 to 0, 255.
  brightness = map(1023 - sensorData, 0, 1023, 0, 255);
  // Use the remapped value as output for the LED.
  analogWrite(ledPin, brightness);

  delay(2);
}
```

https://github.com/expact/arduinolab/blob/master/lab_sensor03/lab_sensor03.ino

Piezo

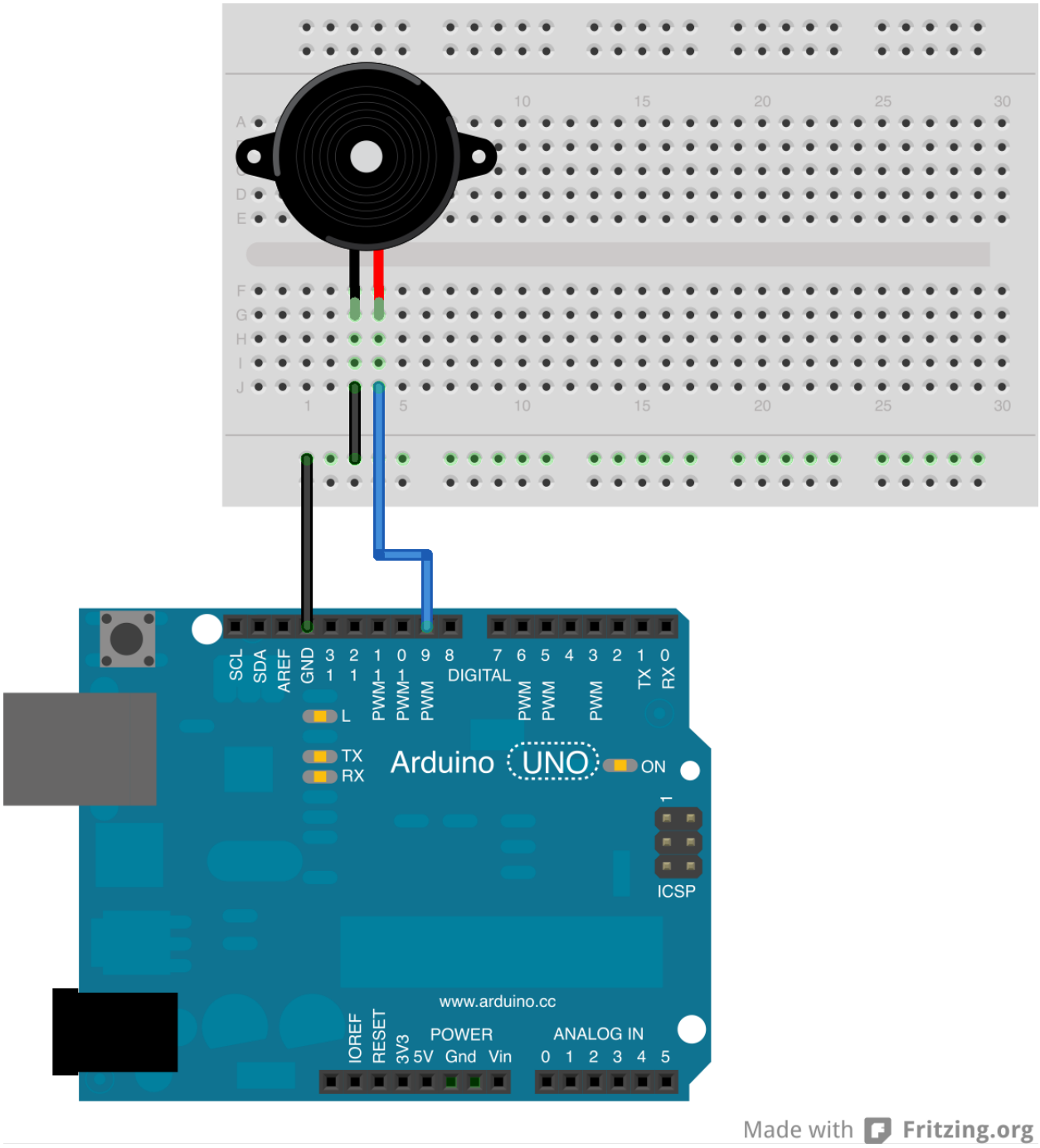
Shows how we can connect a piezo buzzer to make some noise. This demo also shows how can group common tasks in code in methods. See the example of beep.

Components

- Piezo

```
int buzzerPin = 9;

void setup() {
  pinMode(buzzerPin, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  beep(); // Calls the beep method.
  delay(1000);
}

// The beep method, causes the piezo to beep once.
void beep() {
  analogWrite(buzzerPin, 20); // The value here can be anything between 0 and 255,
test a few.
  delay(200); // Test with various values for delay.
  analogWrite(buzzerPin, 0); // This turns of the buzzer.
  delay(200); // Change here to.
}
```

https://github.com/expact/arduinolab/blob/master/lab_piezo01/lab_piezo01.ino



Piezo and a LDR

In this example we link the Piezo togheter with a LDR sensor. Similar to what we previously did with the LED.

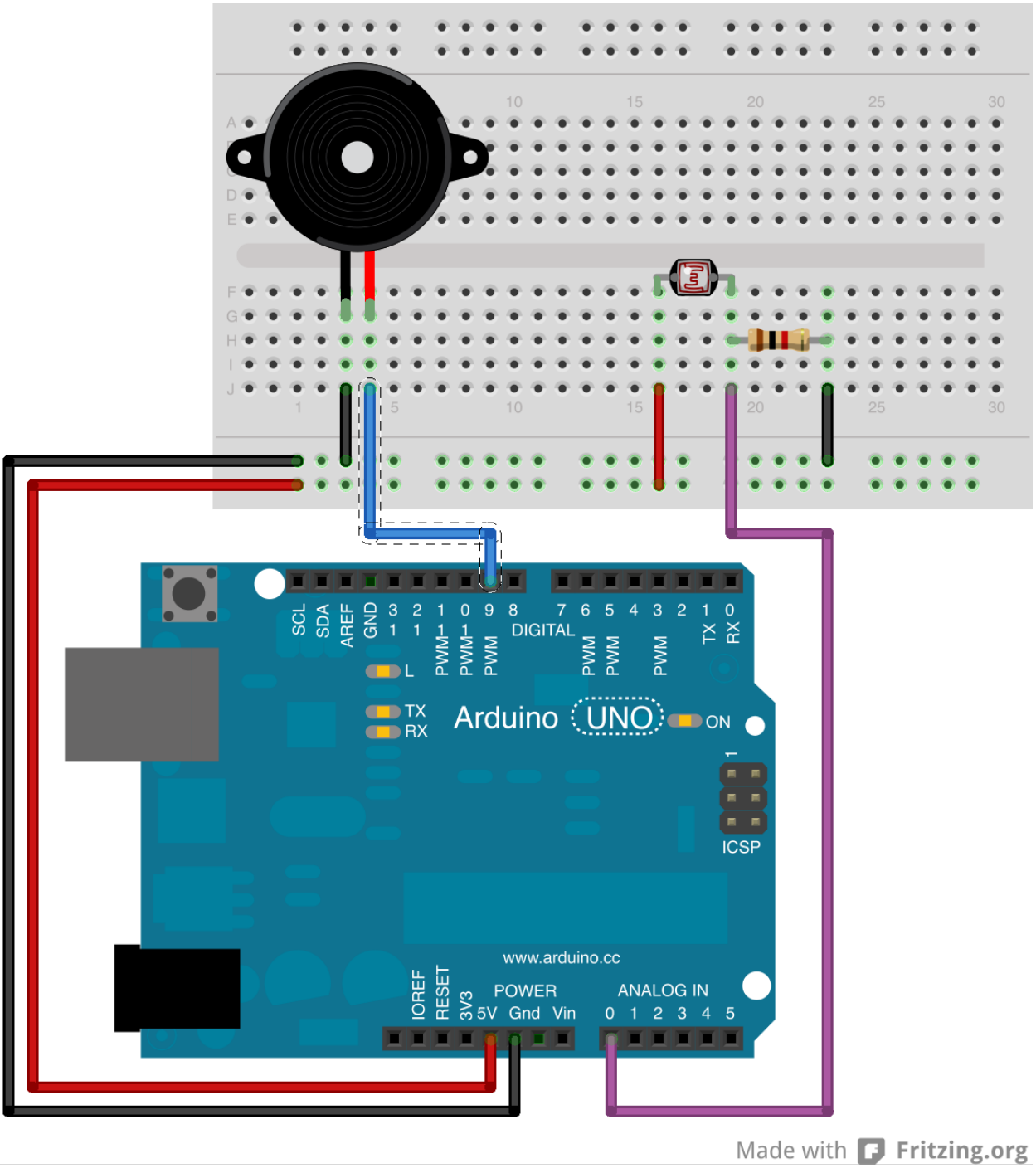
Components

- Piezo

- LDR
- Resistor 1k

```
int buzzerPin = 9; // Digital pin that the buzzer is connected to.
int sensorPin = A0; // Analog pin that the sensor is connected to.

// Initial values for threshold and sensor data.
int sensorData = 0;
int threshold = 950; // Configured threshold.

void setup() {
  pinMode(buzzerPin, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  sensorData = analogRead(sensorPin);
  // Print the dara we recieved from the sensor.
  Serial.println(sensorData);

  // If the sensor value is less than the configured threshold, Beep!
  if (sensorData < threshold) {
    beep(); // Calls the beep method.
  }

  delay(1);
}

// The beep method, causes the piezo to beep once.
void beep() {
  analogWrite(buzzerPin, 20);
  delay(10);
  analogWrite(buzzerPin, 0);
  delay(10);
}
```

https://github.com/expact/arduinolab/blob/master/lab_ldrpiezo/lab_ldrpiezo.ino



Piezo, LDR and a Potentiometer

In our previous example we had a threshold of when to trigger the buzzer. With this demo we connect a potentiometer to use it to configure the threshold.

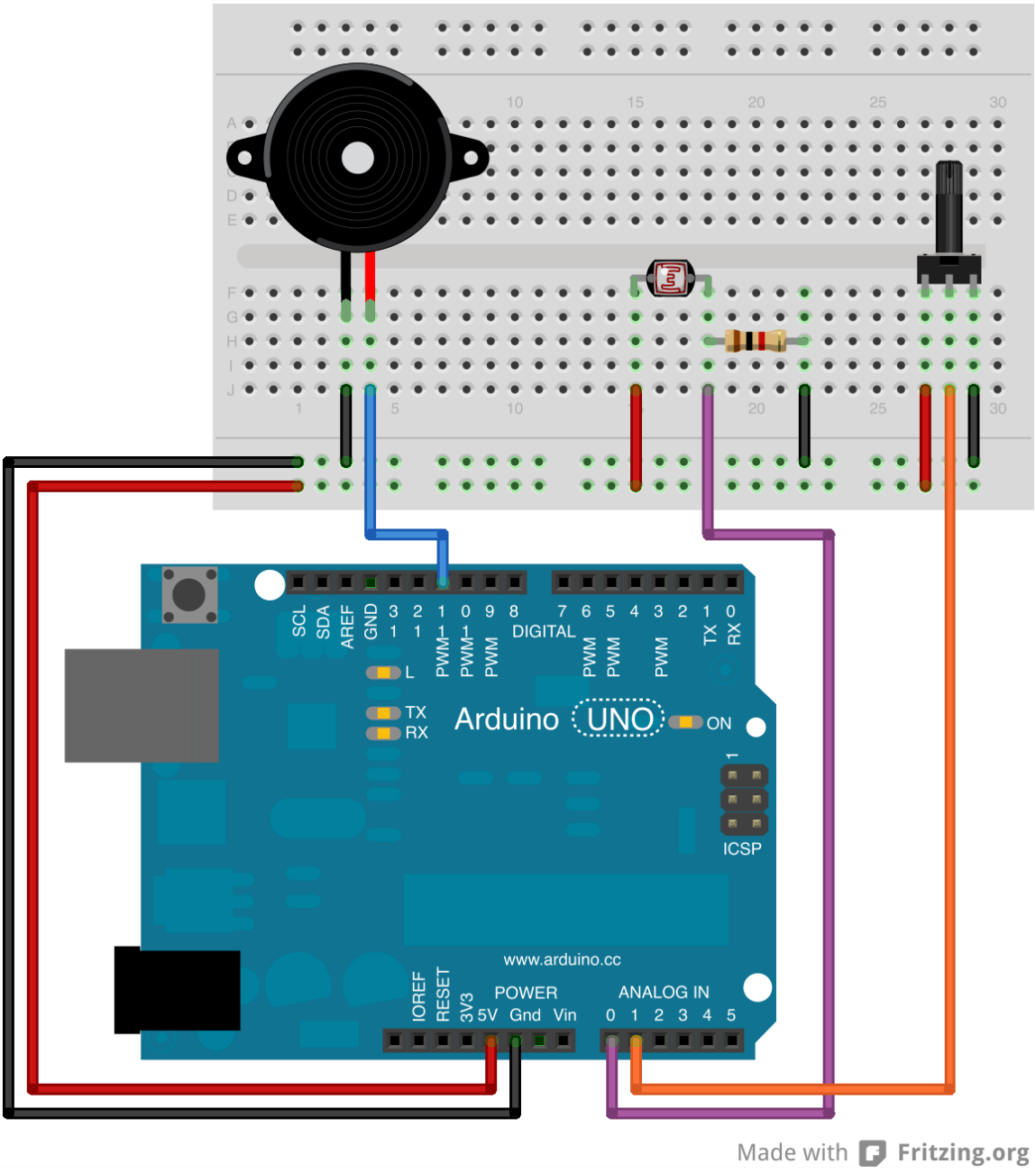
Components

- Piezo

- LDR
- Potentiometer
- Resistor 1k

```
// Building upon the ldrpiezo example we are now using a potentiometer
// to configure the threshhold value.

int buzzerPin = 9; // Digital pin that the buzzer is connected to.
int sensorPin = A0; // Analog pin that the sensor is connected to.
int potPin = A1; // Analog pin that the pot is connected to.

// Initial values for threshold and sensor data.
int sensorData = 0;
int threshold = 0;

void setup() {
  pinMode(buzzerPin, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  sensorData = analogRead(sensorPin);
  threshold = analogRead(potPin); // Read data from the pot into the threshold
variable.

  // Print the data we recieved from the pot.
  Serial.print("Treshhold: ");
  Serial.println(threshold, DEC);

  // If the sensor value is less than the configured threshold, Beep!
  if (sensorData < threshold) {
    beep();
  }

  delay(1);
}

// The beep method, causes the piezo to beep once.
void beep() {
  analogWrite(buzzerPin, 100);
  delay(300);
  analogWrite(buzzerPin, 0);
  delay(300);
}
```

https://github.com/expact/arduinolab/blob/master/lab_ldrpiezopot/lab_ldrpiezopot.ino



Button & Tilt

This demo shows how we can use a button and a tilt sensor as an input.

Components

- LED
- Piezo
- Button

- Tilt
- Resistor 1k
- Resistor 470

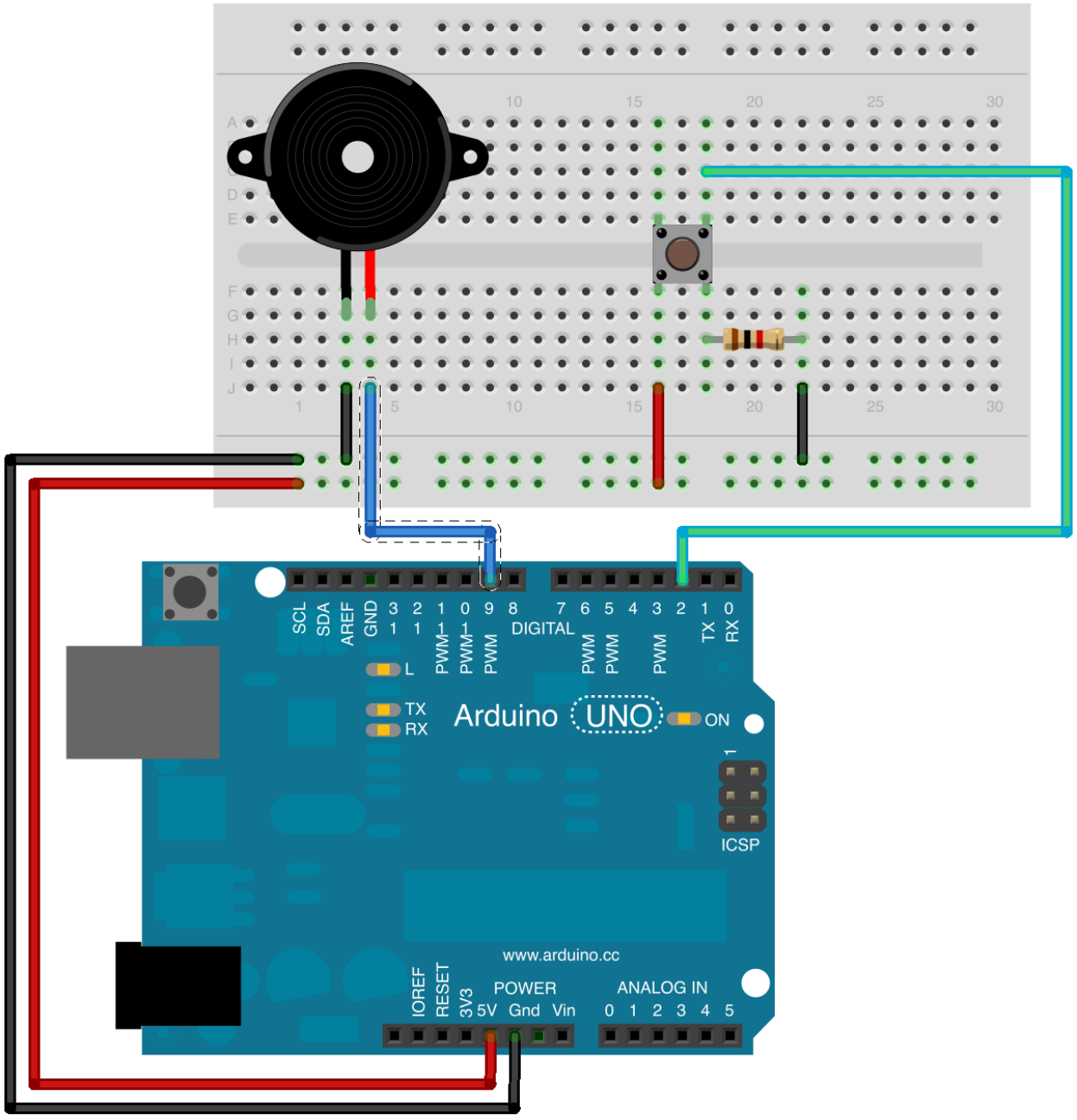
As an exercise change the circuit and the code to turn on and off an LED instead. Refer to the first example of blinking a LED as a reference of how to wire and code a LED to turn on and off.

```
int buzzerPin = 9;
int buttonPin = 2;

int buttonState = 0;

void setup() {
  pinMode(buttonPin, INPUT);
  pinMode(buzzerPin, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  buttonState = digitalRead(buttonPin);

  Serial.println(buttonState);
  if (buttonState == HIGH) {
    beep(); // Calls the beep method.
  }

  delay(2);
}

// The beep method, causes the piezo to beep once.
void beep() {
  analogWrite(buzzerPin, 20);
  delay(10);
  analogWrite(buzzerPin, 0);
  delay(10);
}
```

https://github.com/expact/arduinolab/blob/master/lab_button_piezo01/lab_button_piezo01.ino

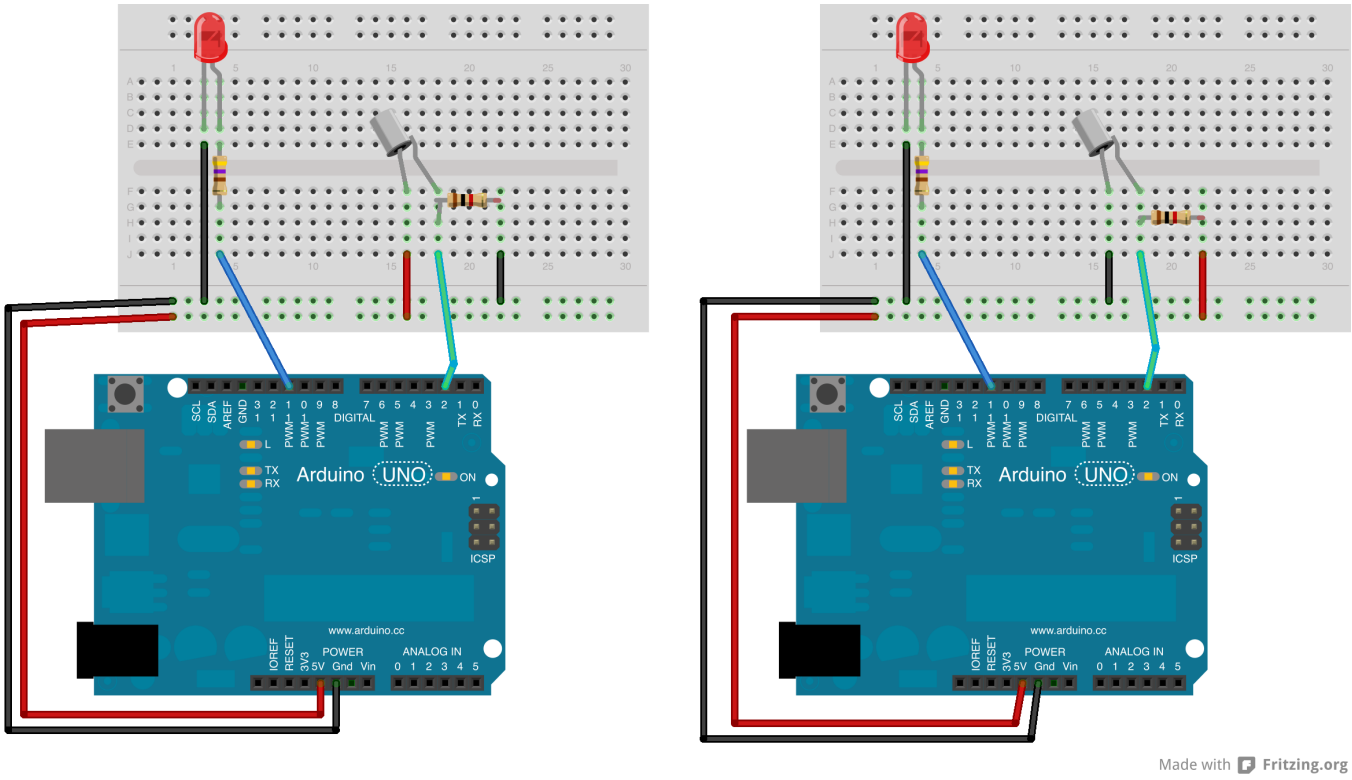
A button example is also provided from the Arduino IDE that is more suited for a LED. It's located at, File -> Examples -> 02 Digital, Button.

Wire the button and upload the code.



Wire the tilt sensor and upload the code.

This is done exactly the same way as the button is wired. The schematic below shows the different between a pull-up resistor (pin to GND, left) and a pull-down resistor (pin to VCC, right).



PIR Sensor

Demonstrates how to wire the PIR sensor. The PIR sensor is used to detect movement.

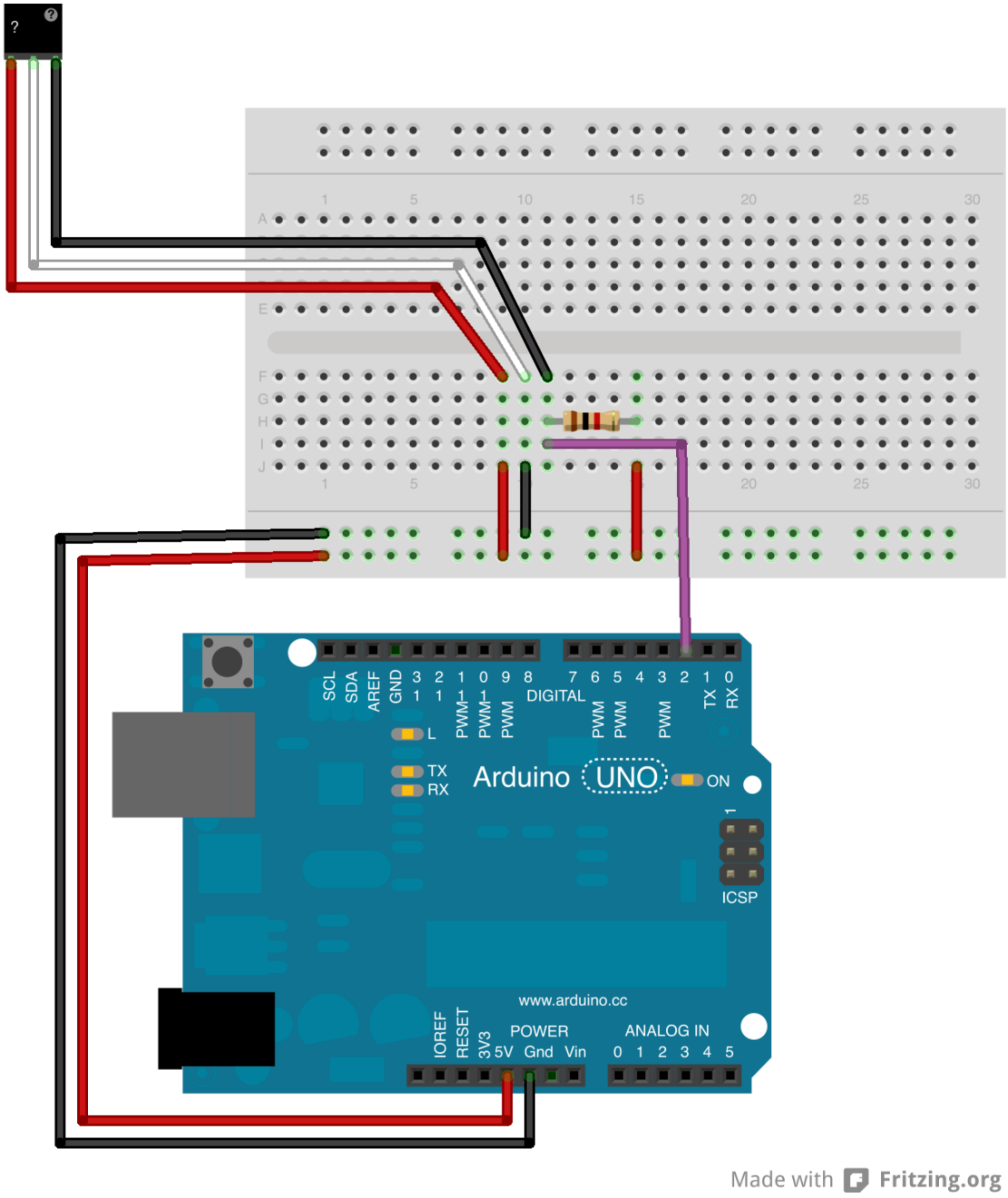
Components

- PIR
- Resistor

```
int pirPin = 2;

void setup(){
 Serial.begin(9600);
 pinMode(pirPin, INPUT);
}

void loop(){
  int pirVal = digitalRead(pirPin);

  if(pirVal == LOW) {
    Serial.println("Motion Detected");
    delay(2000);
  } else {
    Serial.println("No Motion");
    delay(500);
  }
```

```
}
```

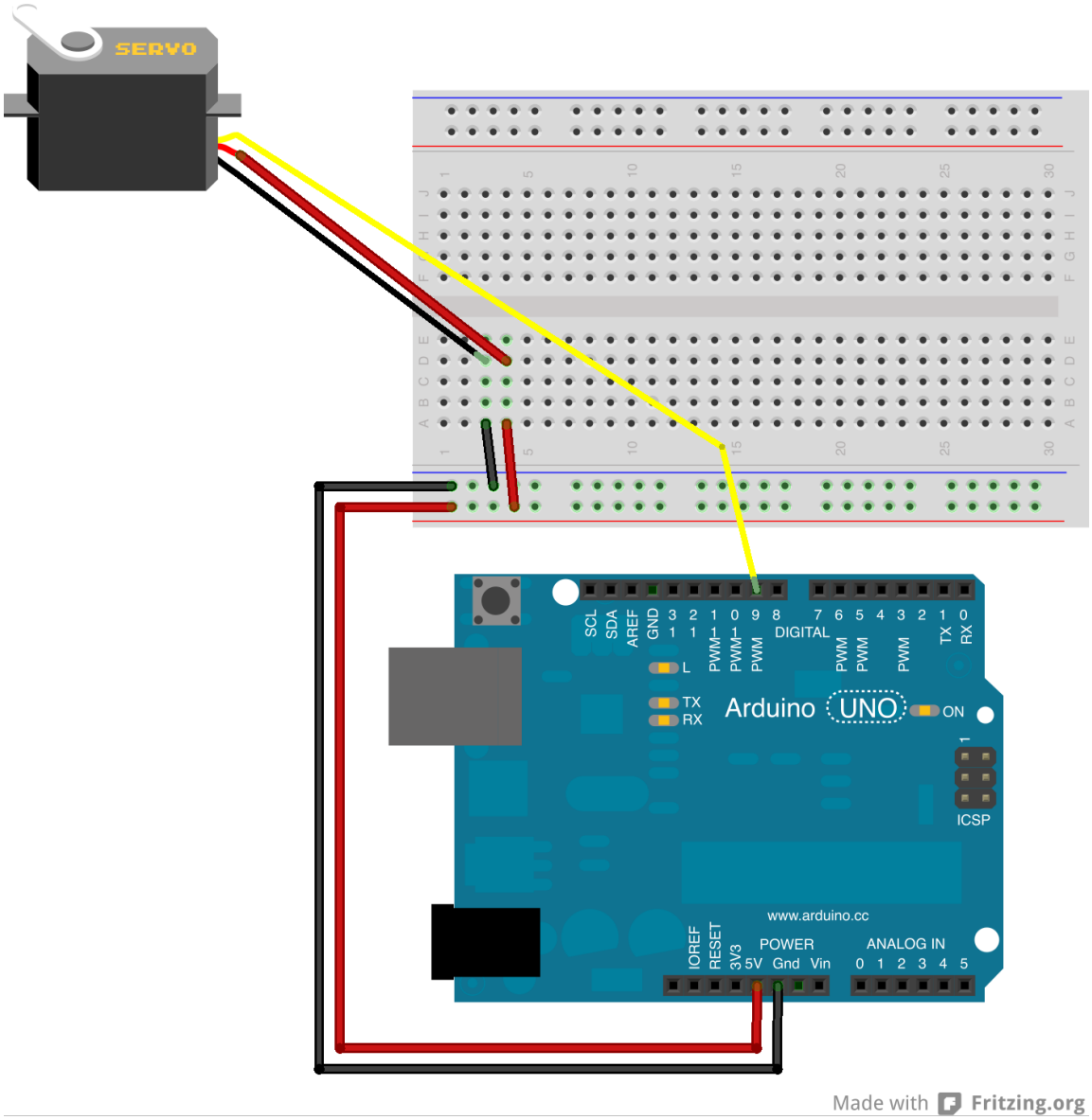
https://github.com/expact/arduinolab/blob/master/lab_pir/lab_pir.ino



Servo

Demonstrates how to wire a servo.

An example code for controlling the servo can be found within the bundled Arduino examples. Go to File, Examples, Servo and choose Sweep.


SERVO
SCL
SDA
AREF
GND
DIGITAL
PWM
L
TX
RX
Arduino
UNO
ON
ICSP
www.arduino.cc
IOREF
RESET
3V3
5V
POWER
Gnd
Vin
ANALOG IN
Made with Fritzing.org