

WEARABLE TECHNOLOGY

Imanol Gómez
imanolgomez.net
29 April 2016



FASHION
HACKDAY

Introduction

Repository

[https://github.com/ImanolGo/
WearableTechnology_FashionHD](https://github.com/ImanolGo/WearableTechnology_FashionHD)

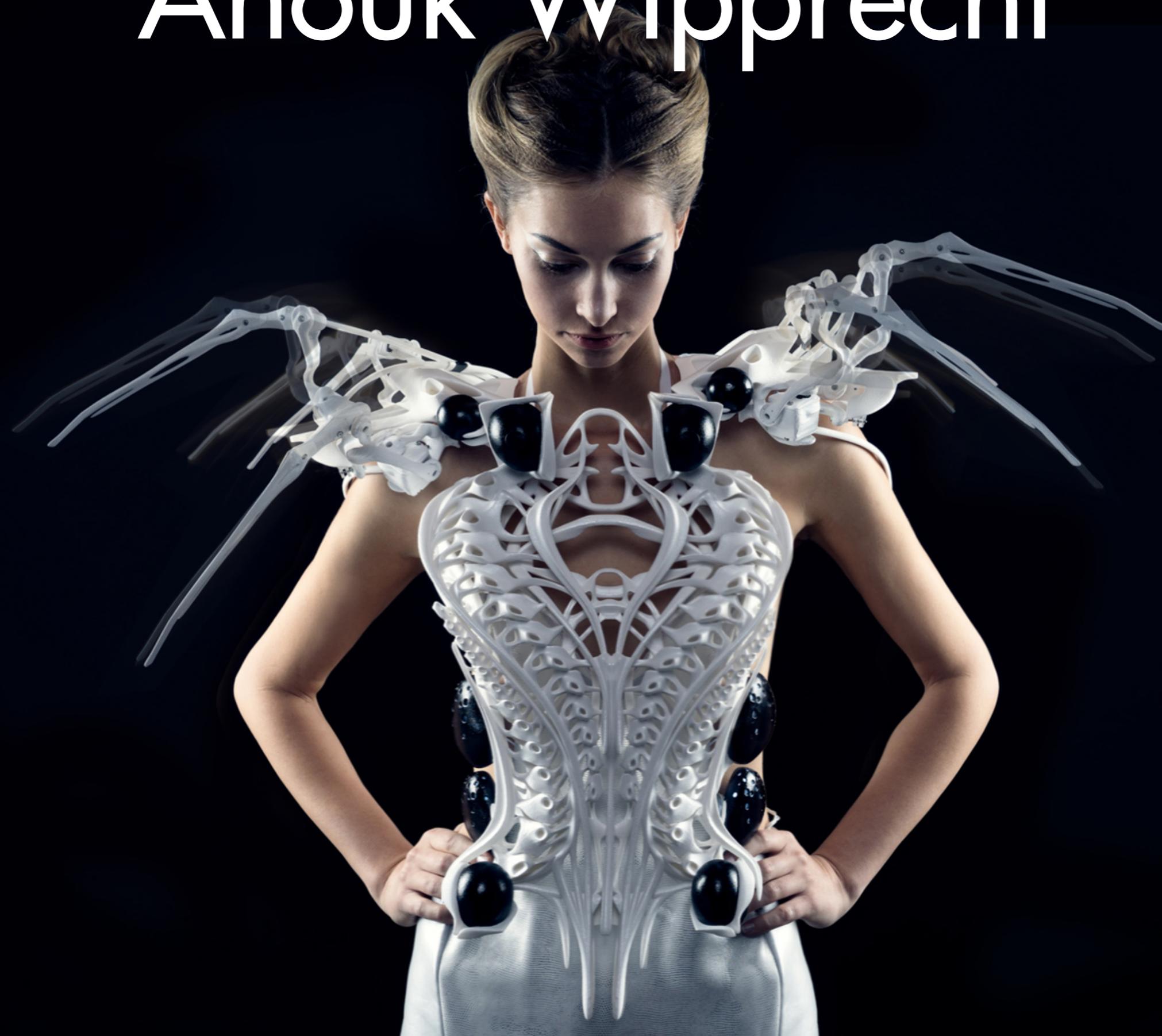


Wearable Solar

Moritz Waldemeyer



Anouk Wipprecht



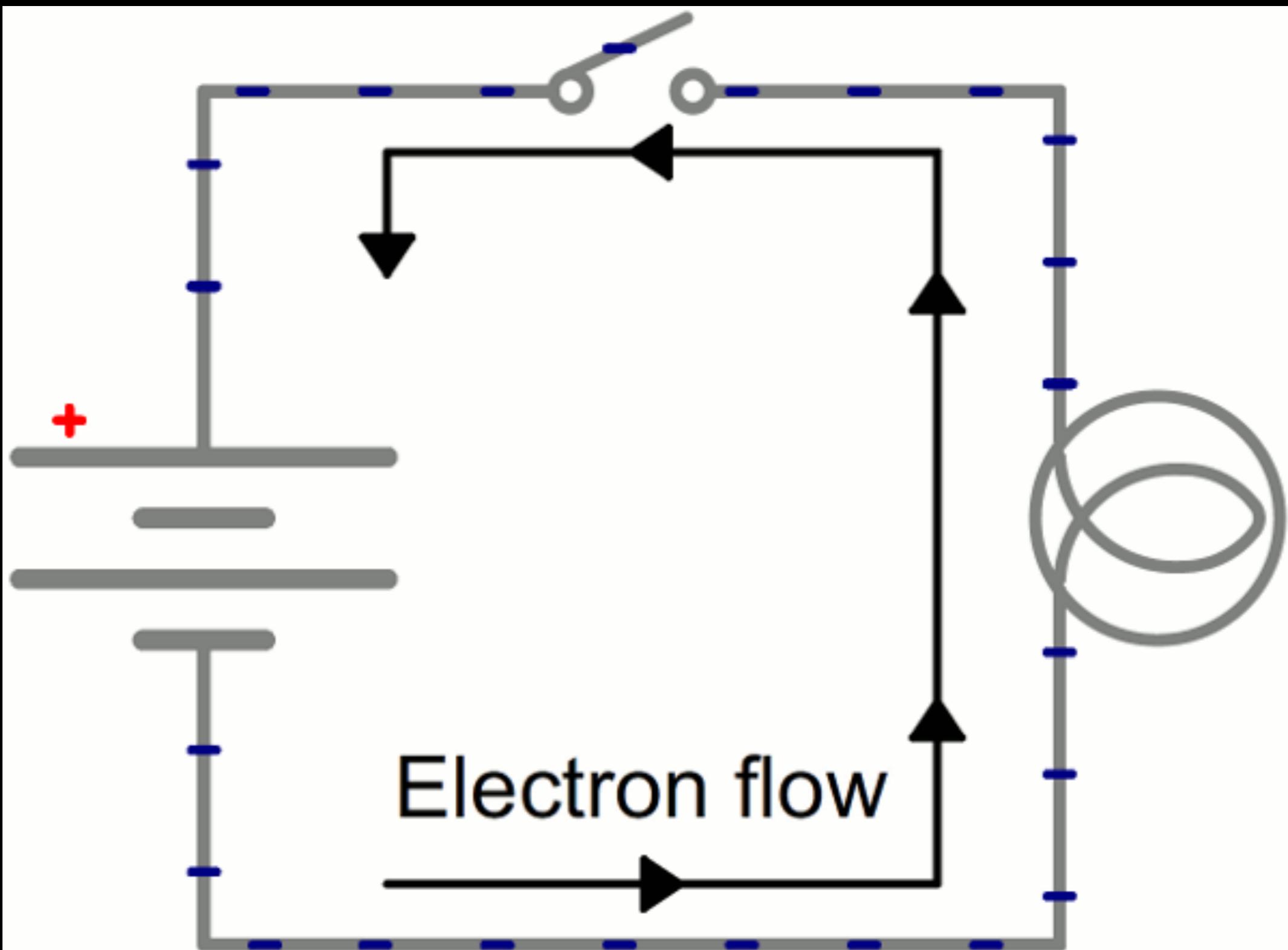
ElektroCouture



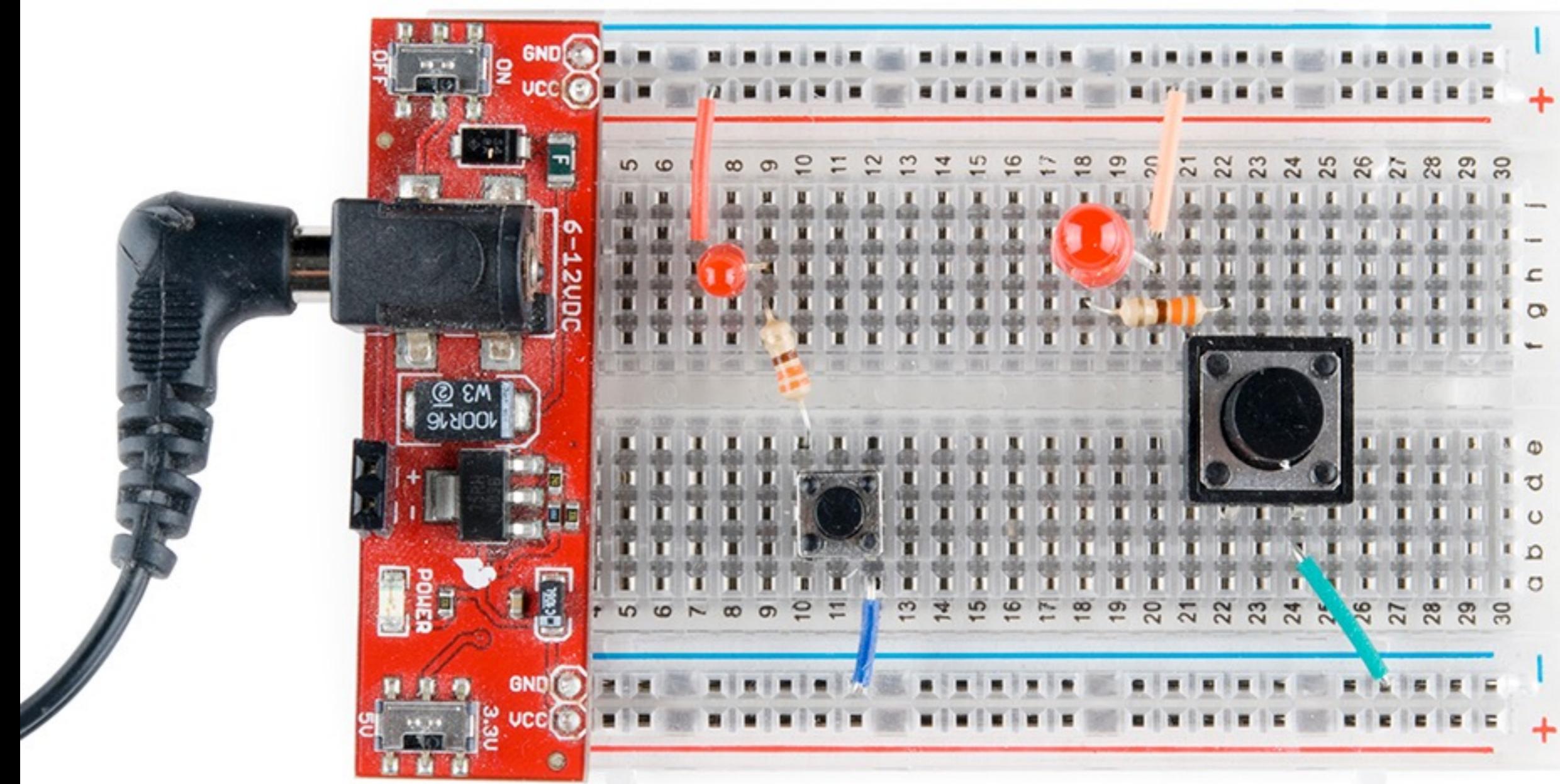
ELEKTROCOUTURE

Circuits

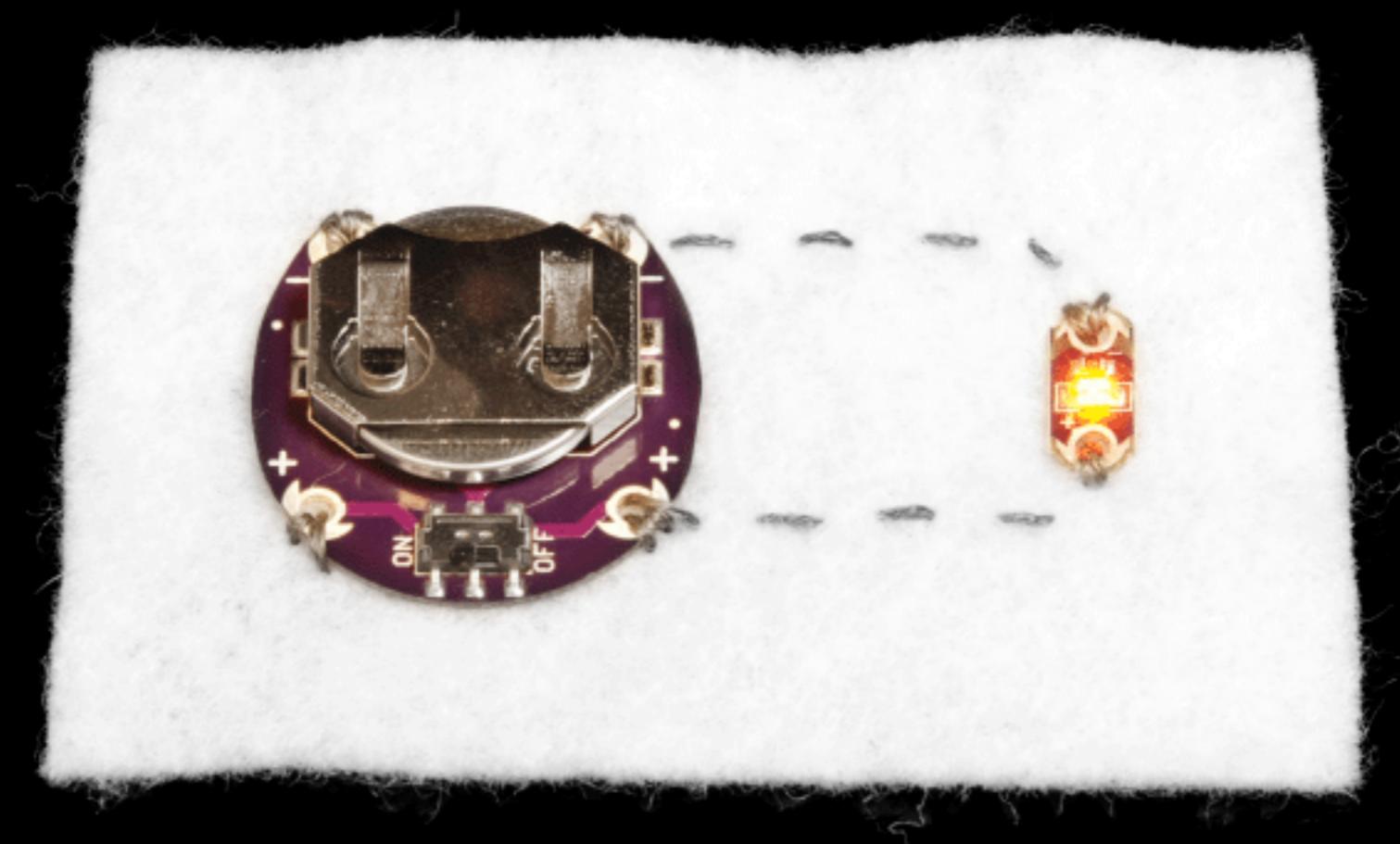
Circuits



Circuits



Circuits



Components

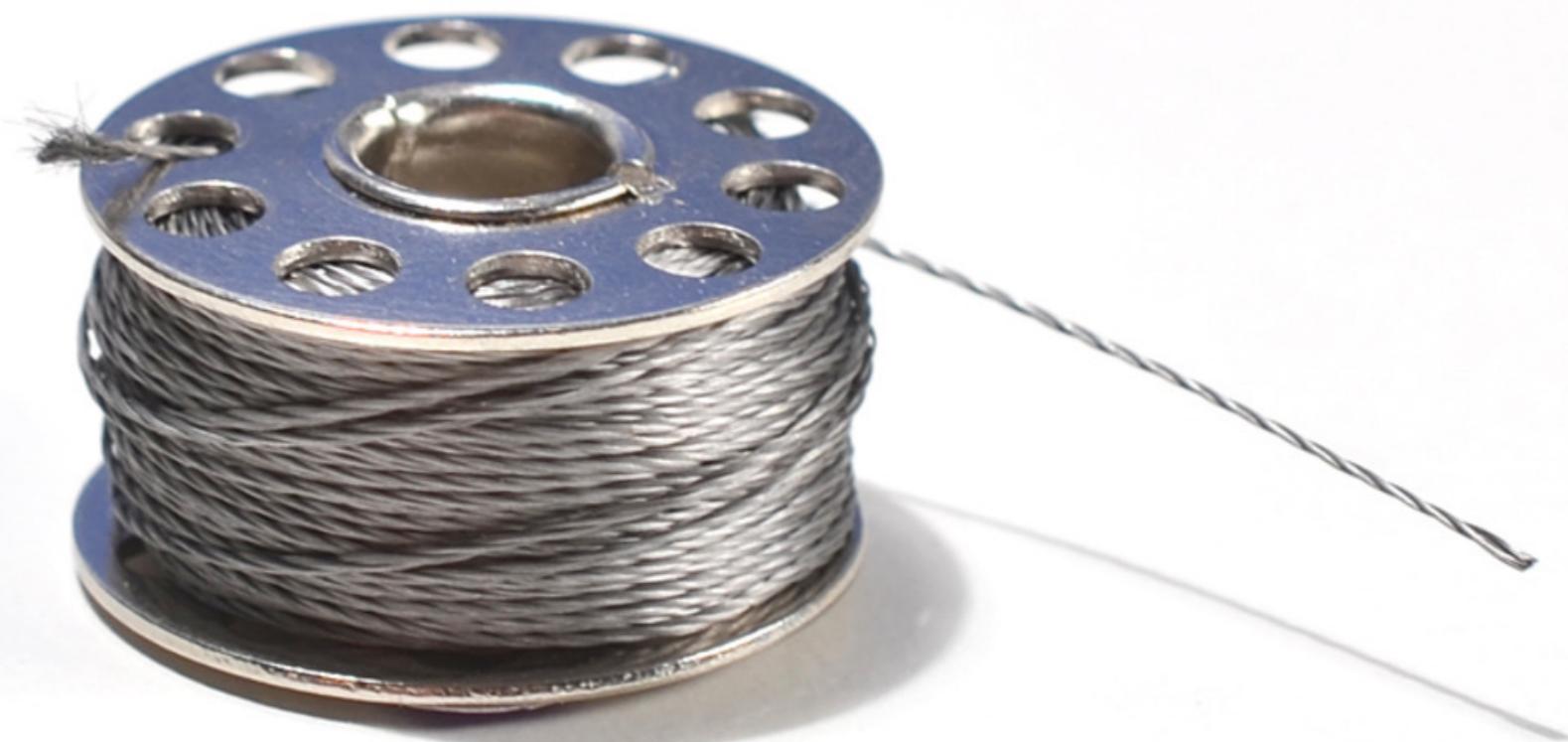
Electric Paint



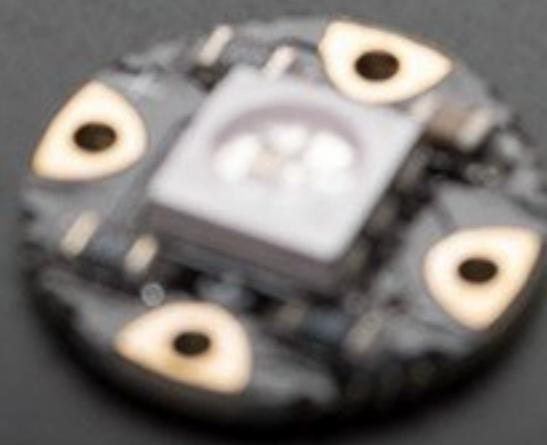
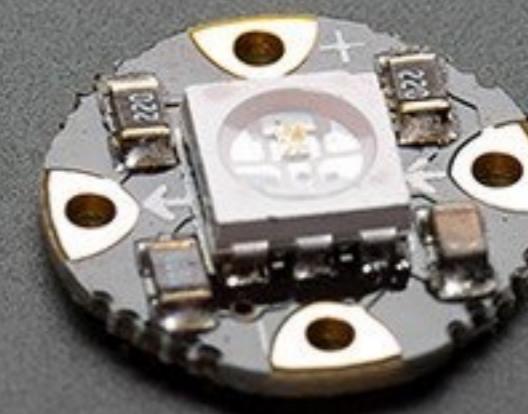
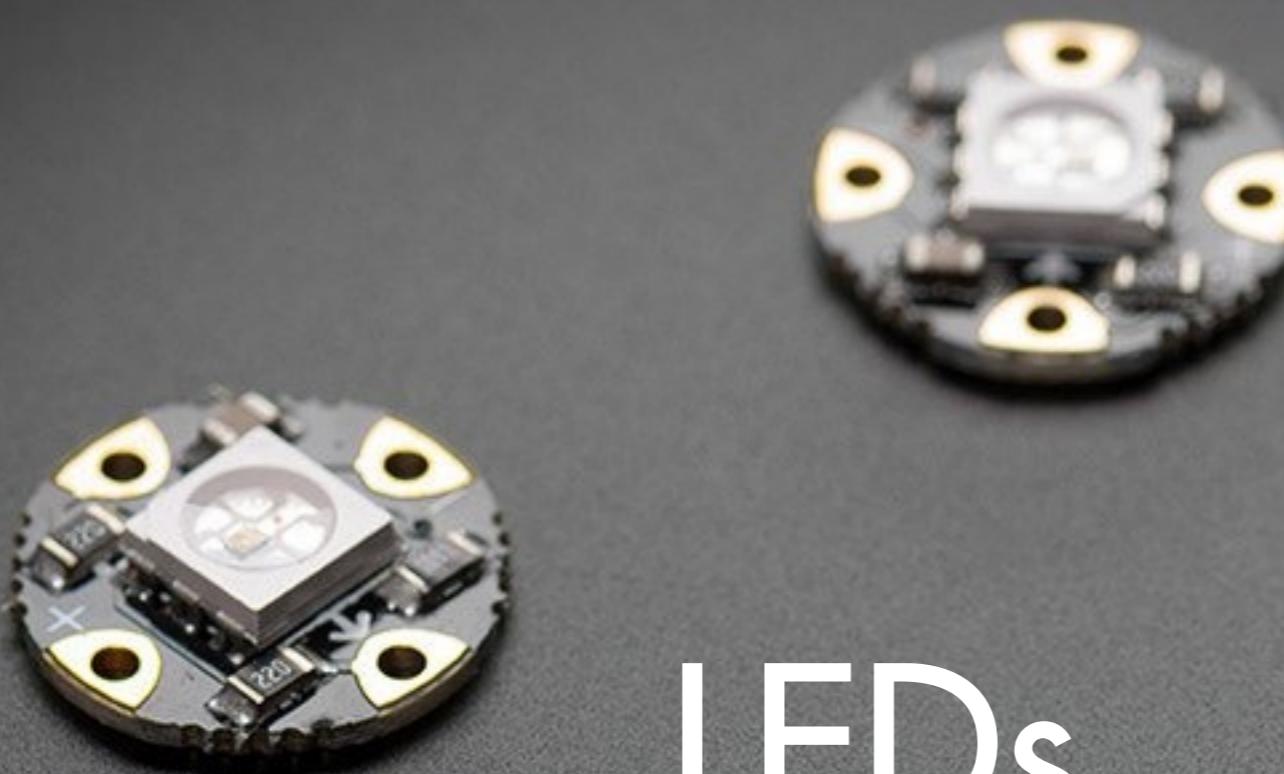
Conductive Fabric



Conductive Thread



LEDs



- LP 503035
500mAh 3.7V
+ 13. 10. 13

LiPo Battery

Sensors

Velostat



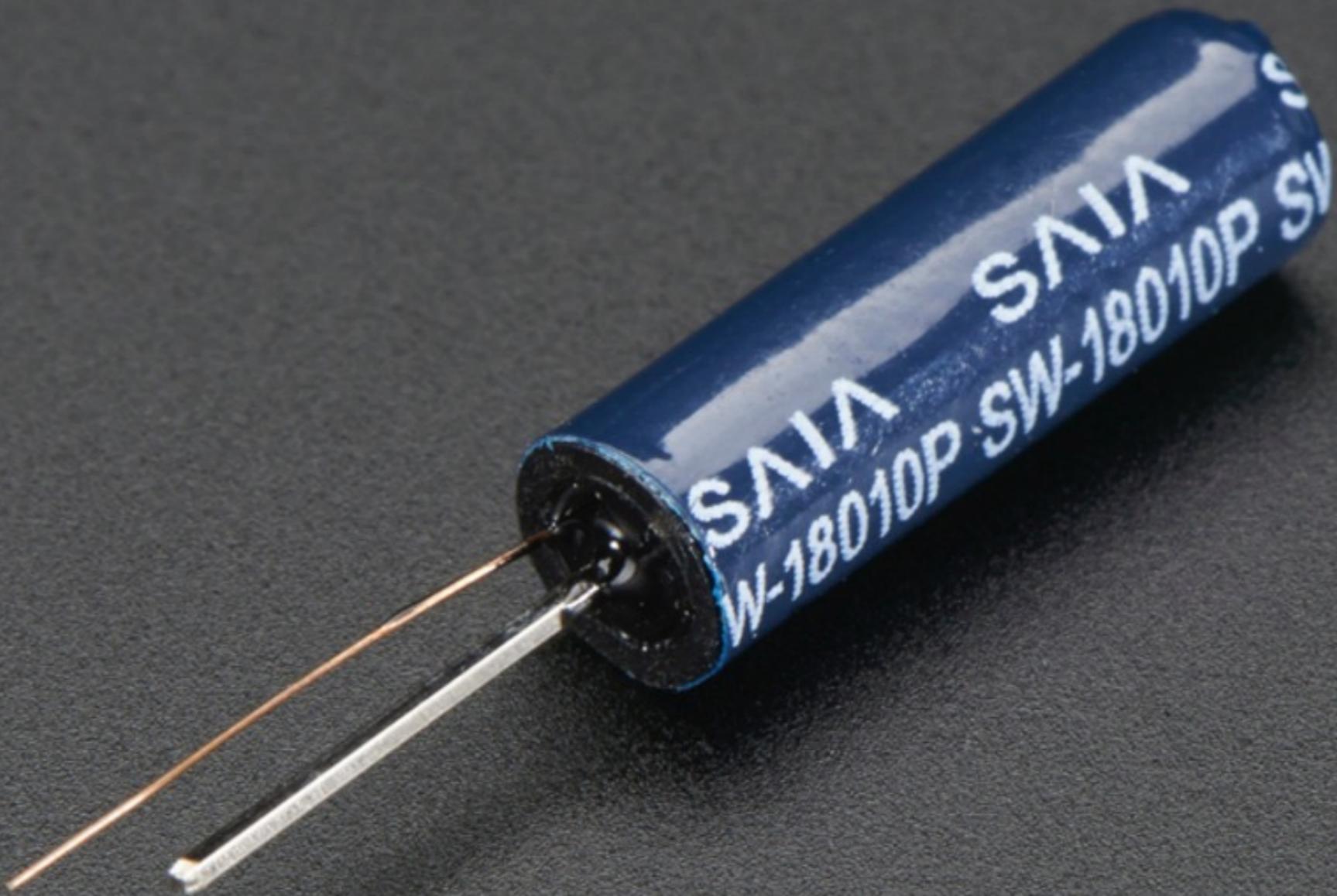
Photoresistor



Switch



Vibration Sensor

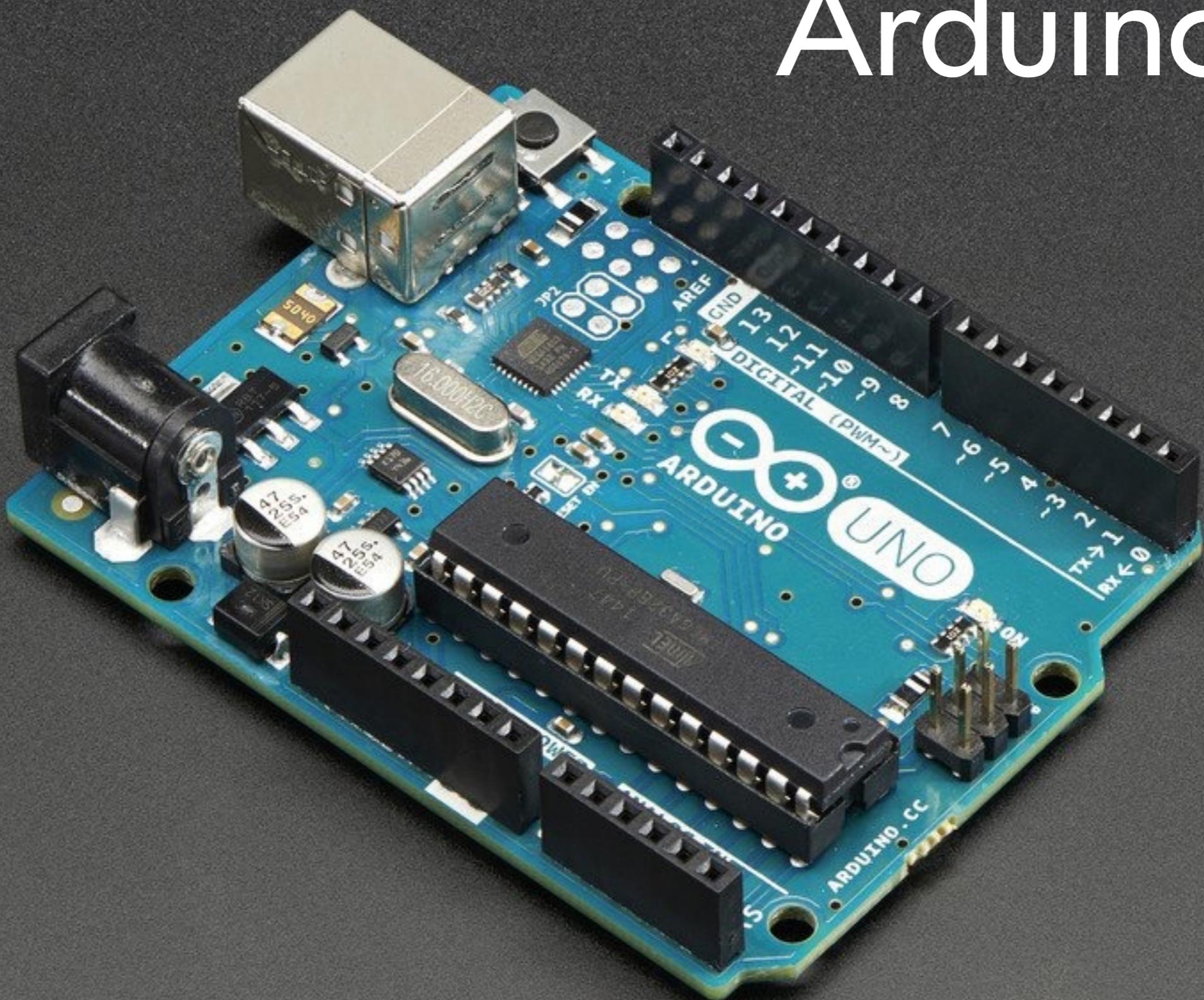




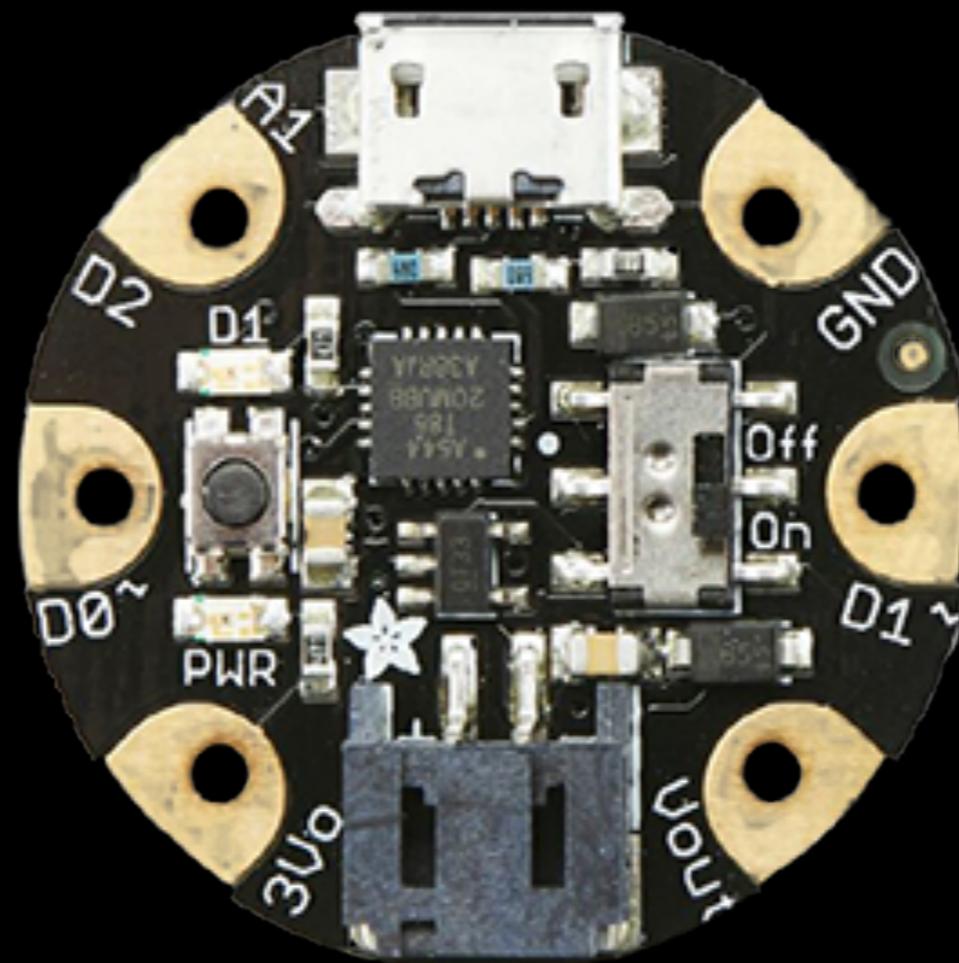
Piezo

Microcontroller

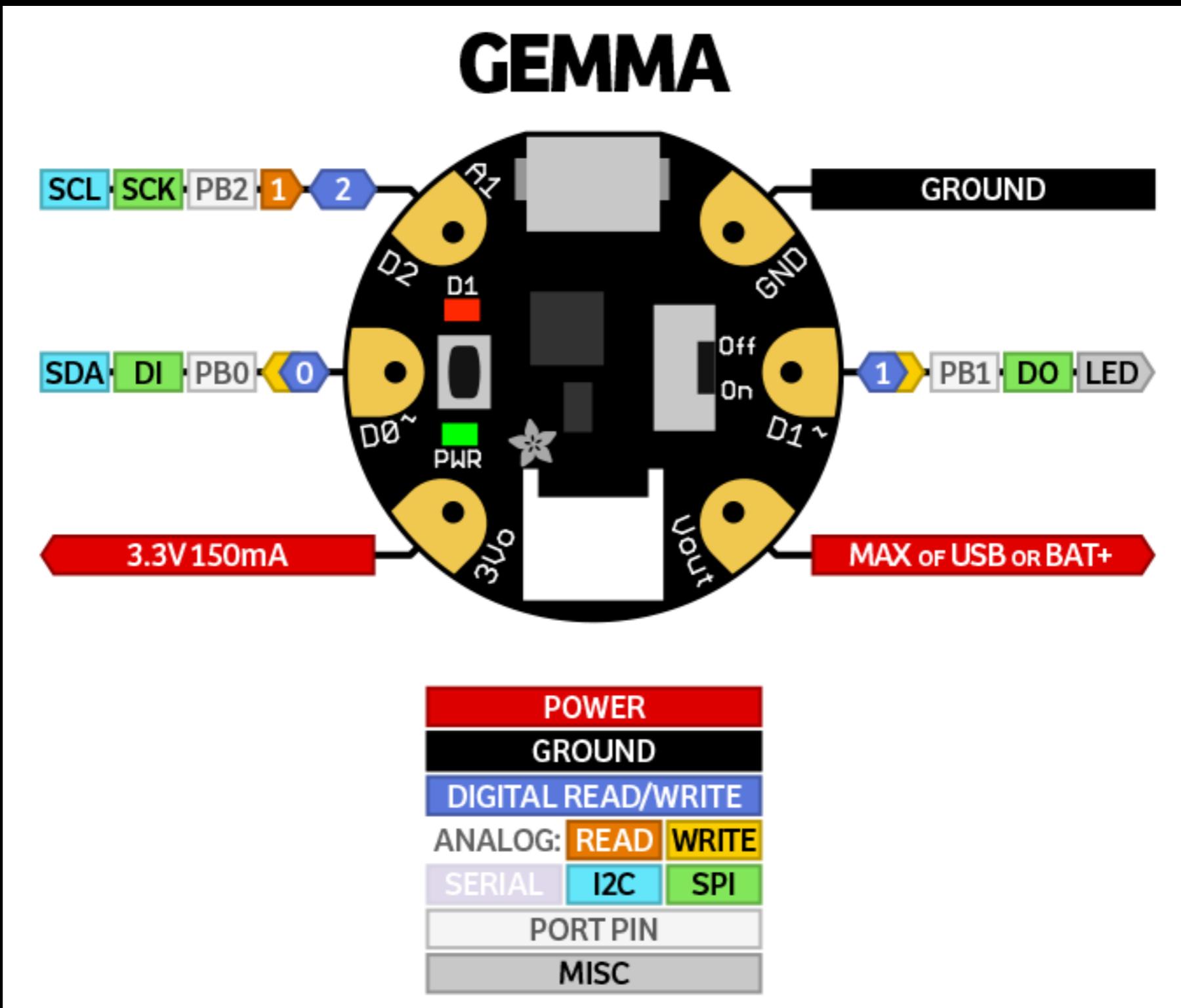
Arduino



Gemma



Pinouts



Arduino IDE



The screenshot shows the Arduino IDE interface with the title bar "Blink | Arduino 1.6.8". The main window displays the "Blink" example sketch. The code is as follows:

```
1 /*
2  * Blink
3  * Turns on an LED on for one second, then off for one second, repeatedly.
4  *
5  * This example code is in the public domain.
6  *
7  * To upload to your or Trinket:
8  * 1) Select the proper board from the Tools->Board Menu (Arduino Gemma if
9  *    teal, Adafruit Gemma if black)
10 * 2) Select the uploader from the Tools->Programmer ("Arduino Gemma" if teal,
11 *    "USBtinyISP" if black Gemma)
12 * 3) Plug in the Gemma into USB, make sure you see the green LED lit
13 * 4) For windows, make sure you install the right Gemma drivers
14 * 5) Press the button on the Gemma/Trinket - verify you see
15 *    the red LED pulse. This means it is ready to receive data
16 * 6) Click the upload button above within 10 seconds
17 */
18
19 int led = 1; // blink 'digital' pin 1 - AKA the built in red LED
20
21 // the setup routine runs once when you press reset:
22 void setup() {
23     // initialize the digital pin as an output.
24     pinMode(led, OUTPUT);
25
26 }
27
28 // the loop routine runs over and over again forever:
29 void loop() {
30     digitalWrite(led, HIGH);
31     delay(1000);
32     digitalWrite(led, LOW);
33     delay(1000);
34 }
```

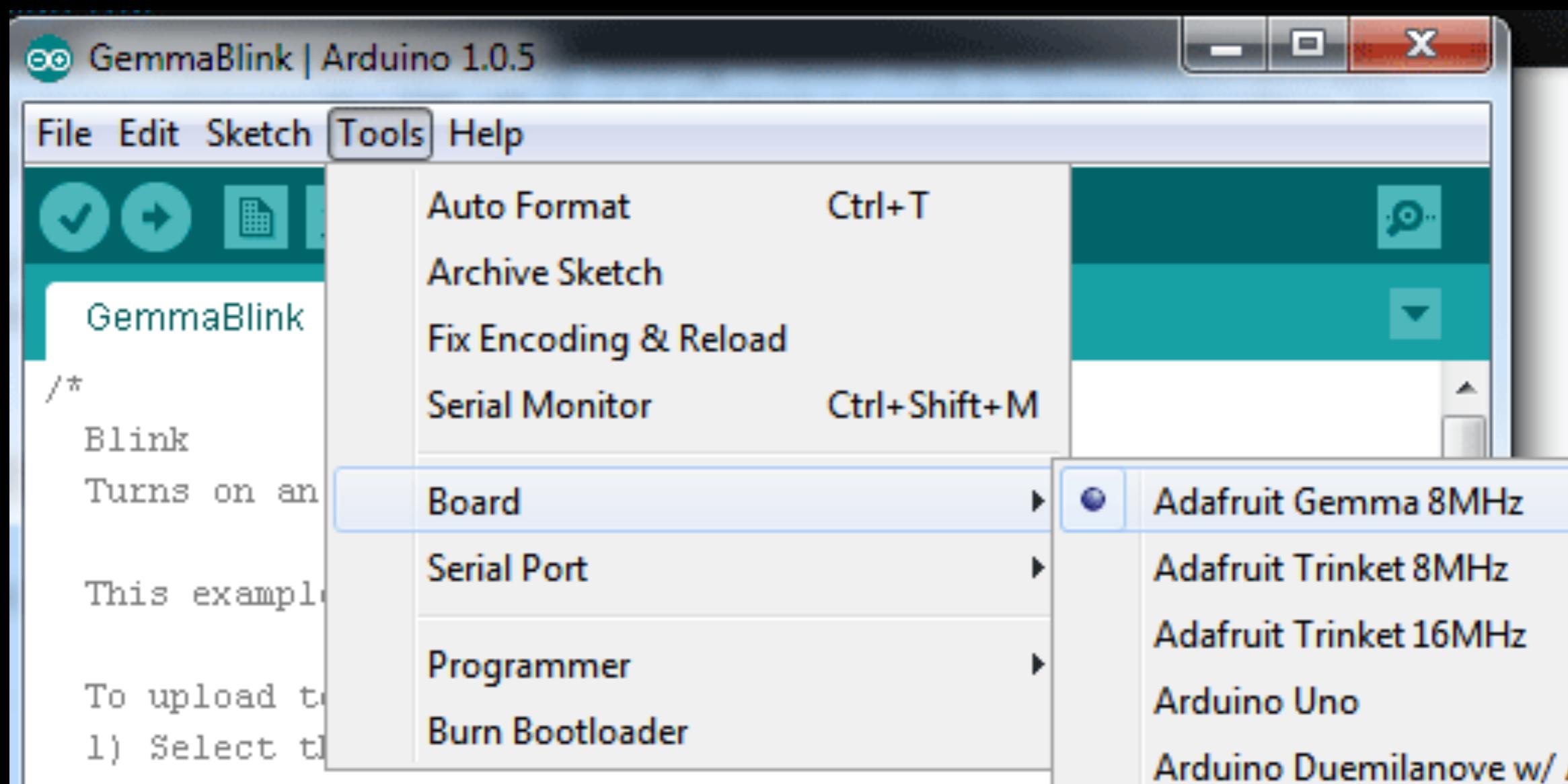
Software Links

- **Windows:** <https://s3.amazonaws.com/adafruit-download/adafruit-arduino-1.6.4-windows.zip>
- **Mac OS X:** <https://s3.amazonaws.com/adafruit-download/adafruit-arduino-1.6.4-macosx.zip>
- **Linux:** <https://s3.amazonaws.com/adafruit-download/adafruit-arduino-1.6.4-linux32.tar.xz>

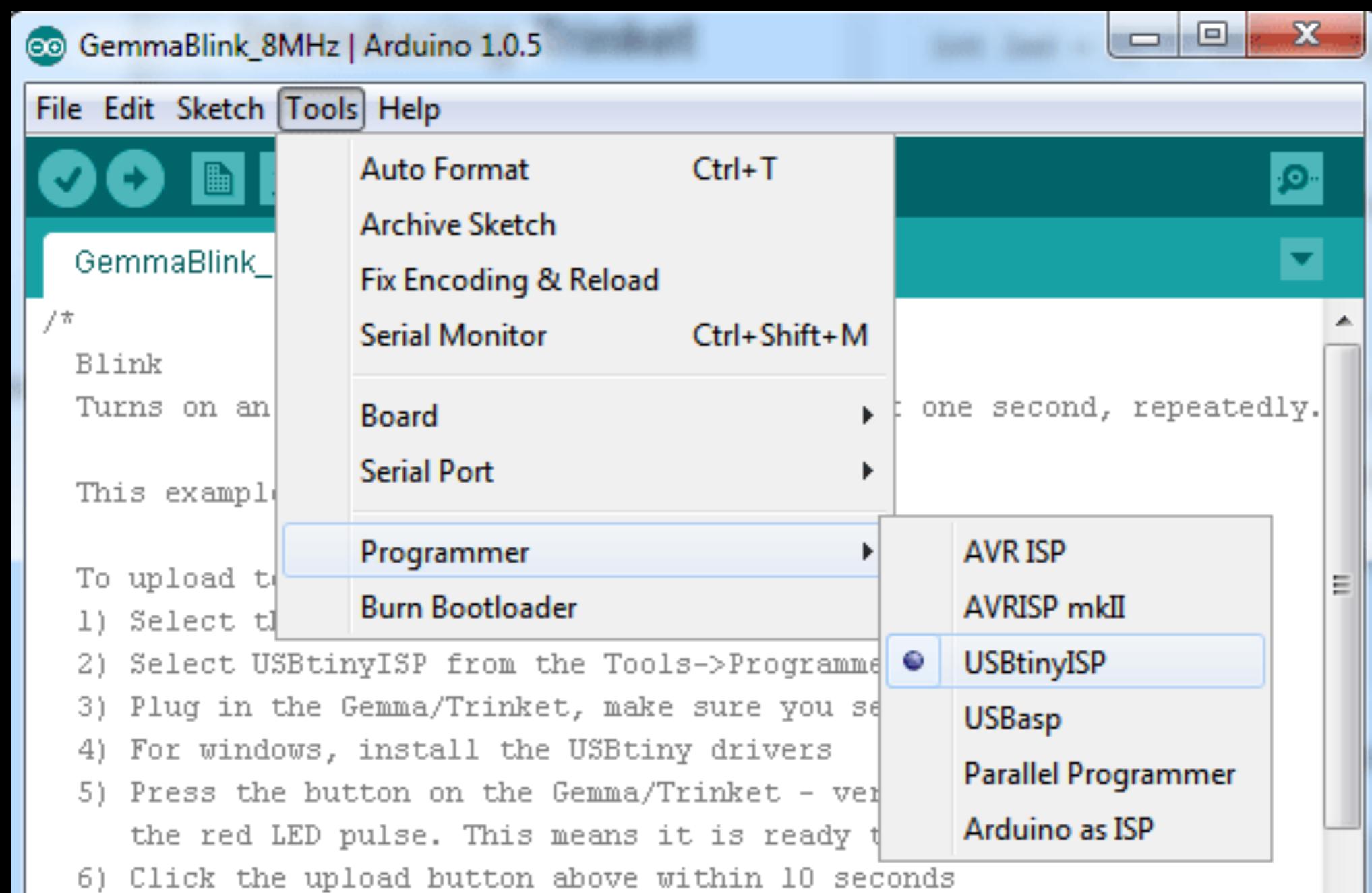
Windows Driver

[https://github.com/adafruit/Adafruit_Windows_Drivers/
releases/download/1.0.0.0/adafruit_drivers.exe](https://github.com/adafruit/Adafruit_Windows_Drivers/releases/download/1.0.0.0/adafruit_drivers.exe)

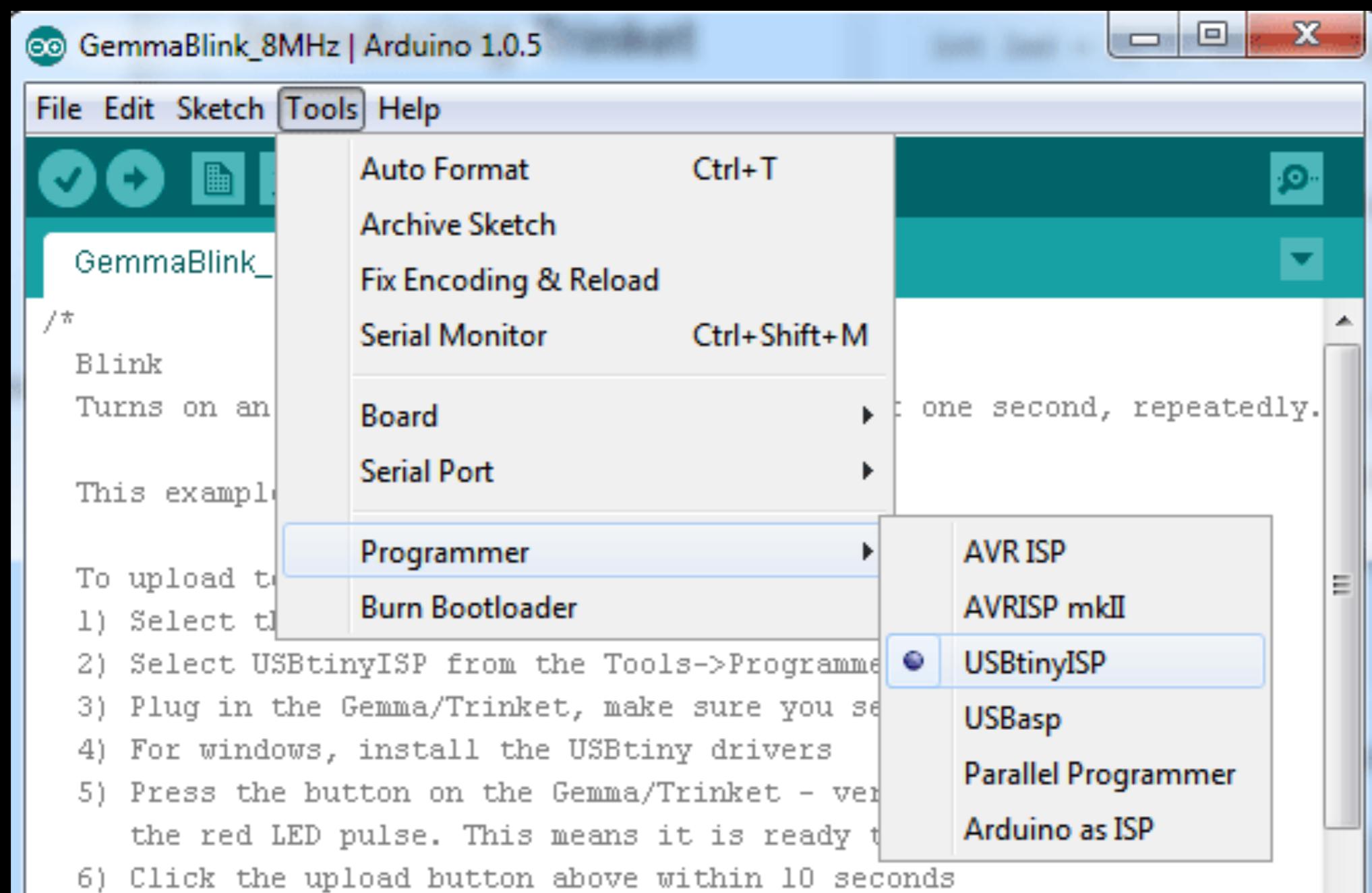
Select the board



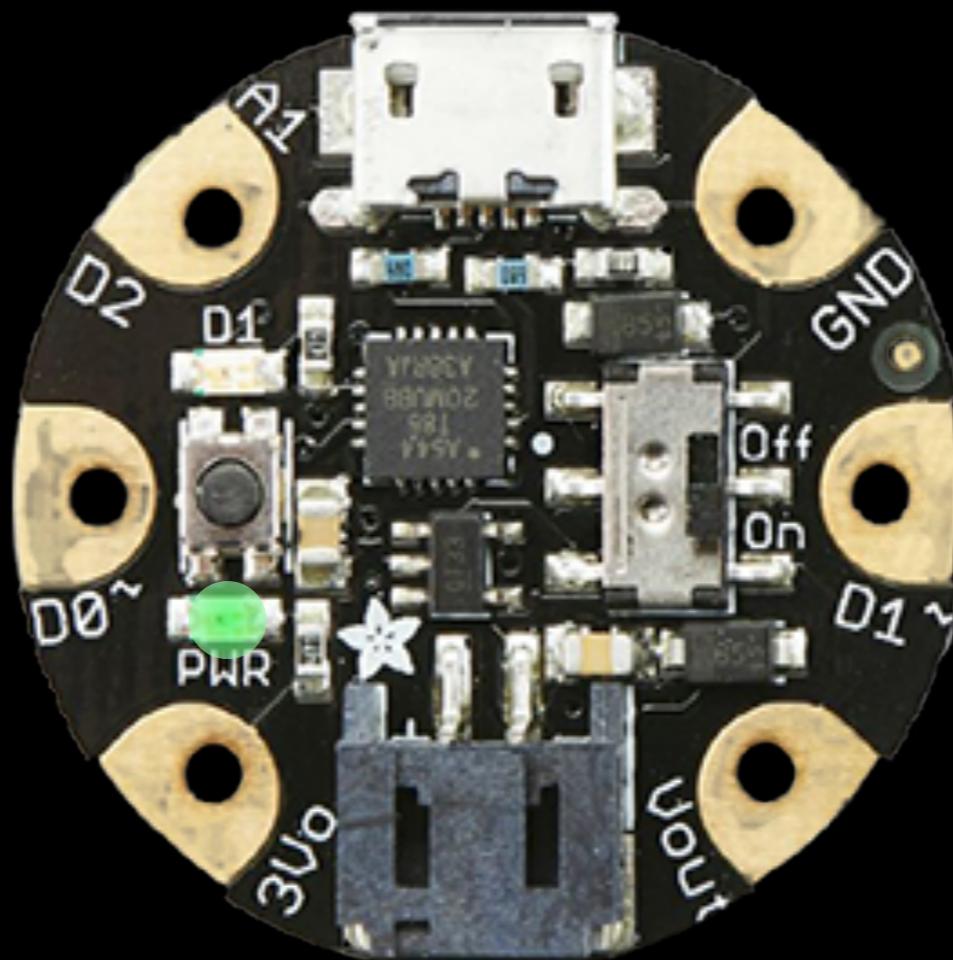
Select the programmer



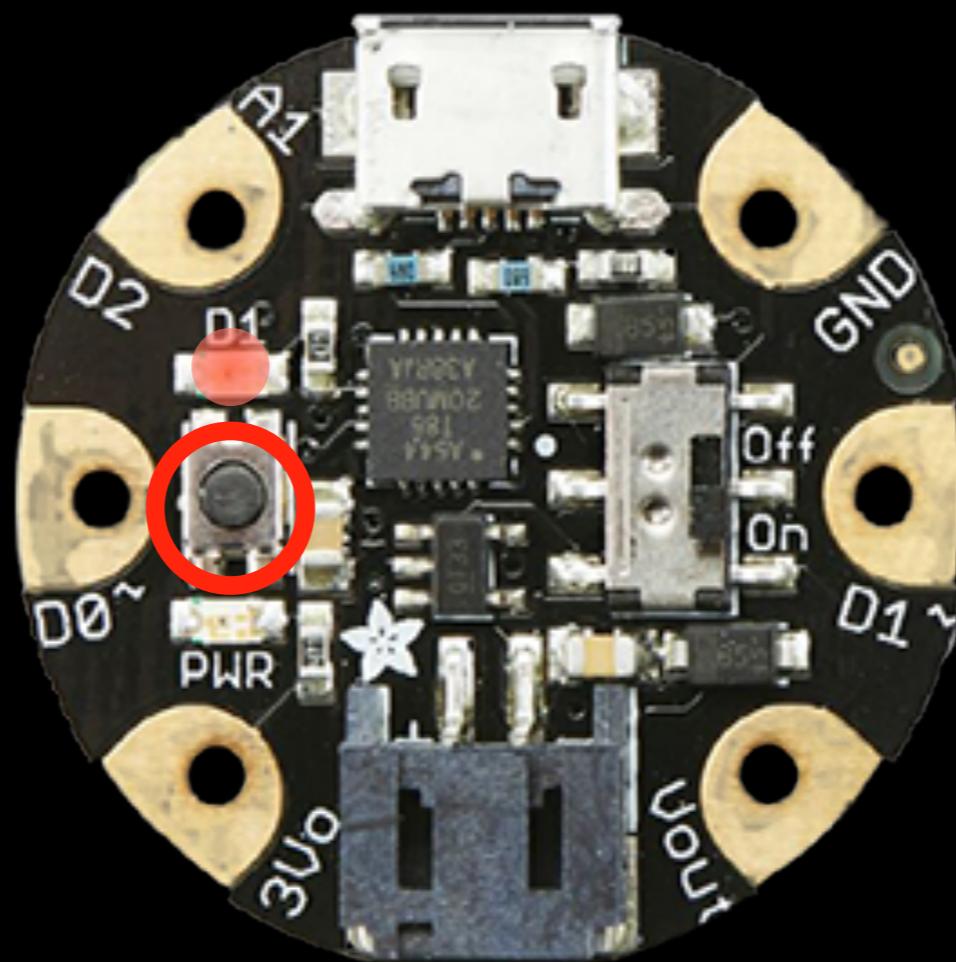
Select the programmer



Plug the board



Plug the board



Upload



Examples

Blink



The screenshot shows the Arduino IDE interface with the title bar "Blink | Arduino 1.6.8". Below the title bar is a toolbar with icons for file operations (New, Open, Save, Upload, Download) and a search function. The main window displays the "Blink" example sketch. The code is as follows:

```
1 /*  
2  * Blink  
3  * Turns on an LED on for one second, then off for one second, repeatedly.  
4  *  
5  * This example code is in the public domain.  
6  *  
7  * To upload to your or Trinket:  
8  * 1) Select the proper board from the Tools->Board Menu (Arduino Gemma if teal, Adafruit Gemma if black)  
9  * 2) Select the uploader from the Tools->Programmer ("Arduino Gemma" if teal,  
10 * "USBtinyISP" if black Gemma)  
11 * 3) Plug in the Gemma into USB, make sure you see the green LED lit  
12 * 4) For windows, make sure you install the right Gemma drivers  
13 * 5) Press the button on the Gemma/Trinket - verify you see  
14 *     the red LED pulse. This means it is ready to receive data  
15 * 6) Click the upload button above within 10 seconds  
16 */  
17  
18 int led = 1; // blink 'digital' pin 1 - AKA the built in red LED  
19  
20 // the setup routine runs once when you press reset:  
21 void setup() {  
22     // initialize the digital pin as an output.  
23     pinMode(led, OUTPUT);  
24  
25 }  
26  
27  
28 // the loop routine runs over and over again forever:  
29 void loop() {  
30     digitalWrite(led, HIGH);  
31     delay(1000);  
32     digitalWrite(led, LOW);  
33     delay(1000);  
34 }
```

Button



The screenshot shows the Arduino IDE interface with the title bar "Button | Arduino 1.6.8". The main window displays the "Button" example sketch. The code is as follows:

```
1 /*  
2  * Button  
3  * Turns on an LED when a switch connected from #0 to ground is pressed  
4  *  
5  * This example code is in the public domain.  
6  *  
7  * To upload to your Gemma or Trinket:  
8  * 1) Select the proper board from the Tools->Board Menu  
9  * 2) Select USBtinyISP from the Tools->Programmer  
10 * 3) Plug in the Gemma/Trinket, make sure you see the green LED lit  
11 * 4) For windows, install the USBtiny drivers  
12 * 5) Press the button on the Gemma/Trinket - verify you see  
13 *     the red LED pulse. This means it is ready to receive data  
14 * 6) Click the upload button above within 10 seconds  
15 */  
16  
17 #define SWITCH 0  
18 #define LED 1  
19  
20 // the setup routine runs once when you press reset:  
21 void setup() {  
22     // initialize the LED pin as an output.  
23     pinMode(LED, OUTPUT);  
24     // initialize the SWITCH pin as an input.  
25     pinMode(SWITCH, INPUT);  
26     // ...with a pullup  
27     digitalWrite(SWITCH, HIGH);  
28 }  
29  
30 // the loop routine runs over and over again forever:  
31 void loop() {  
32     if (! digitalRead(SWITCH)) { // if the button is pressed  
33         digitalWrite(LED, HIGH);    // light up the LED  
34     } else {  
35         digitalWrite(LED, LOW);   // otherwise, turn it off  
36     }  
37 }
```

Pulse



The screenshot shows the Arduino IDE interface with the title bar "Pulse | Arduino 1.6.8". The central area displays the "Pulse" sketch code. The code demonstrates how to pulse the internal LED on a Gemma or Trinket using the `analogWrite` function.

```
1 /*  
2  * Pulse  
3  * Pulses the internal LED to demonstrate the analogWrite function  
4  *  
5  * This example code is in the public domain.  
6  *  
7  * To upload to your Gemma or Trinket:  
8  * 1) Select the proper board from the Tools->Board Menu  
9  * 2) Select USBtinyISP from the Tools->Programmer  
10 * 3) Plug in the Gemma/Trinket, make sure you see the green LED lit  
11 * 4) For windows, install the USBtiny drivers  
12 * 5) Press the button on the Gemma/Trinket - verify you see  
13 *     the red LED pulse. This means it is ready to receive data  
14 * 6) Click the upload button above within 10 seconds  
15 */  
16  
17 int led = 1; // pulse 'digital' pin 1 - AKA the built in red LED  
18  
19 // the setup routine runs once when you press reset:  
20 void setup() {  
21     // initialize the digital pin as an output.  
22     pinMode(led, OUTPUT);  
23 }  
24  
25 // the loop routine runs over and over again forever:  
26 void loop() {  
27     for (int i=0; i<256; i++) {  
28         analogWrite(led, i); // PWM the LED from 0 to 255 (max)  
29         delay(5);  
30     }  
31     for (int i=255; i>=0; i--) {  
32         analogWrite(led, i); // PWM the LED from 255 (max) to 0  
33         delay(5);  
34     }  
35 }
```

Resources

- <http://www.kobakant.at/DIY/>
- <https://learn.adafruit.com/category/flora>
- <https://learn.sparkfun.com/tutorials/tags/wearables?page=all>

Questions?

Imanol Gómez

imanolgomez.net

yo@imanolgomez.net