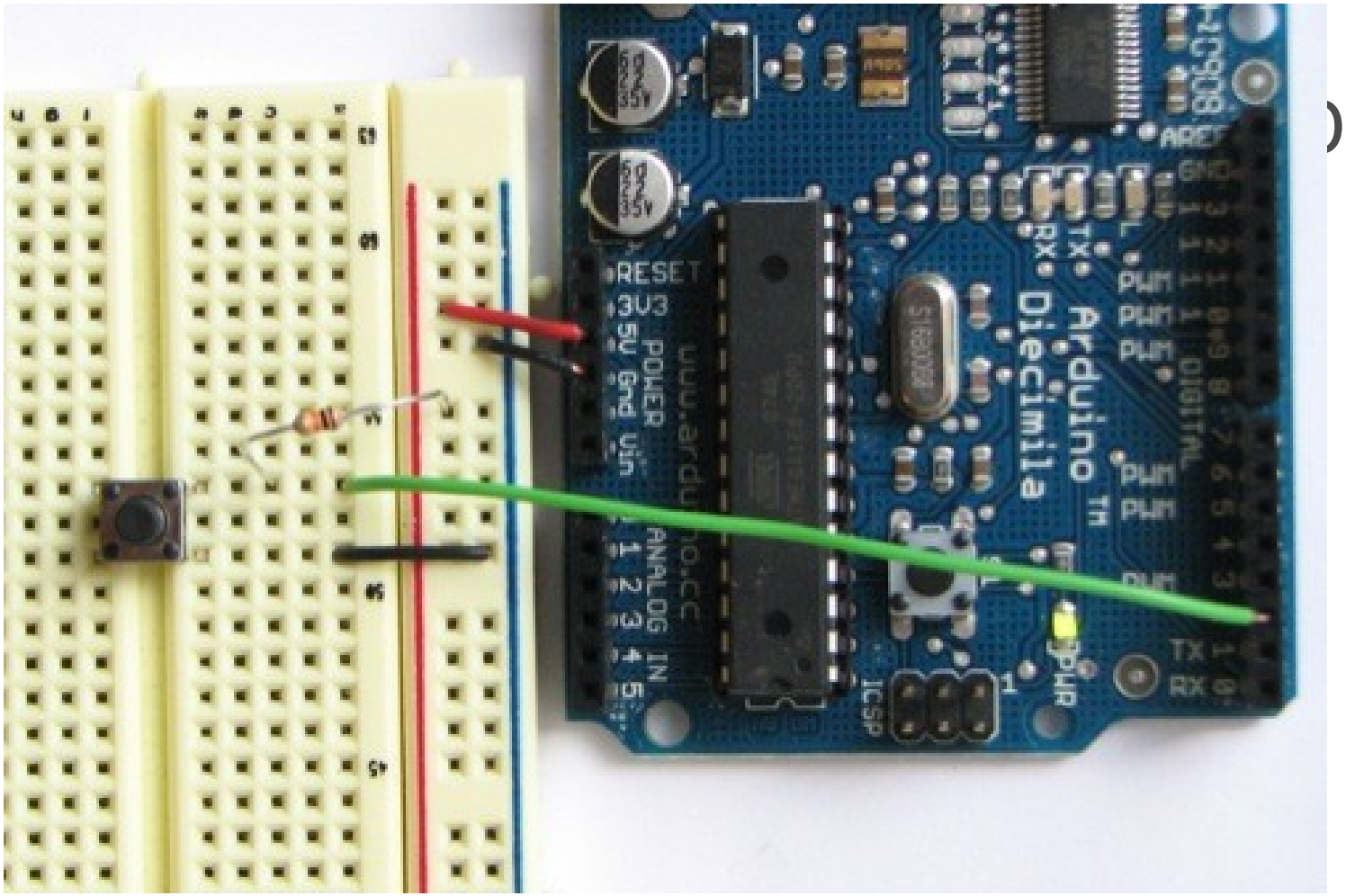
Do-It-Yourself

} buttons and sensors {


RESET
3V3
5V
GND
GND
Vin
POWER
ANALOG IN
Arduino
Diecimila
TX
RX
PWR
ICSP
DIGITAL
PWM

Make your own switch

Find this example in:

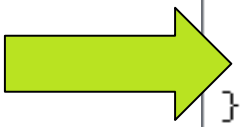
File > Examples > Digital > Button

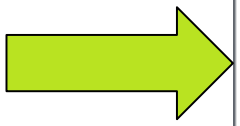
http://www.arduino.cc/en/Tutorial/Button

Turns on and off a light emitting diode(LED) connected to
digital pin 13, when pressing a pushbutton attached to pin 2.

The circuit:
* LED attached from pin 13 to ground
* pushbutton attached to pin 2 from +5V
* 10K resistor attached to pin 2 from ground

```
// 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

// variables will change:
int buttonState = 0;         // variable for reading the pushbutton status

void setup() {
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);
}

void loop(){
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);

  // check if the pushbutton is pressed.
  // if it is, the buttonState is HIGH:
  if (buttonState == HIGH) {
    // turn LED on:
    digitalWrite(ledPin, HIGH);
  }
  else {
    // turn LED off:
    digitalWrite(ledPin, LOW);
  }
}
```

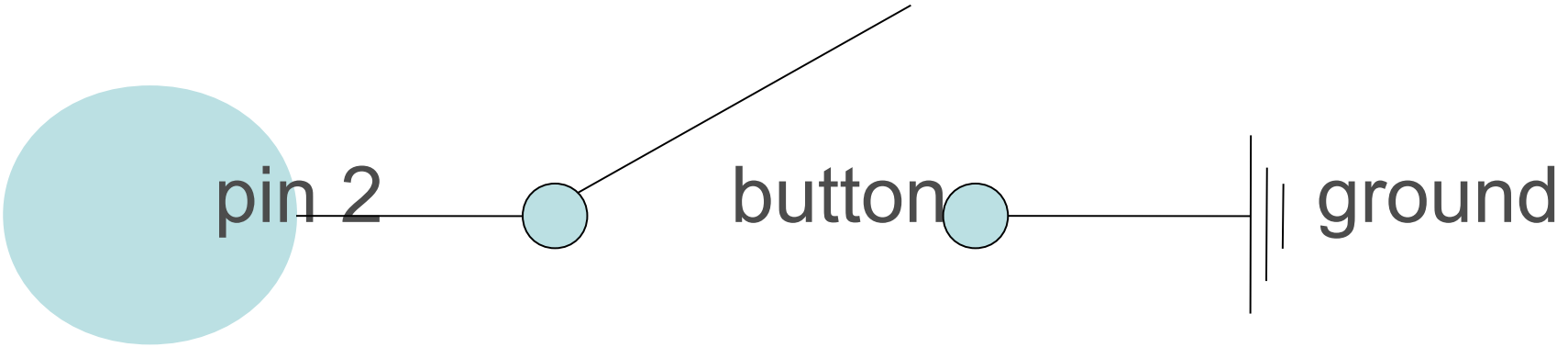


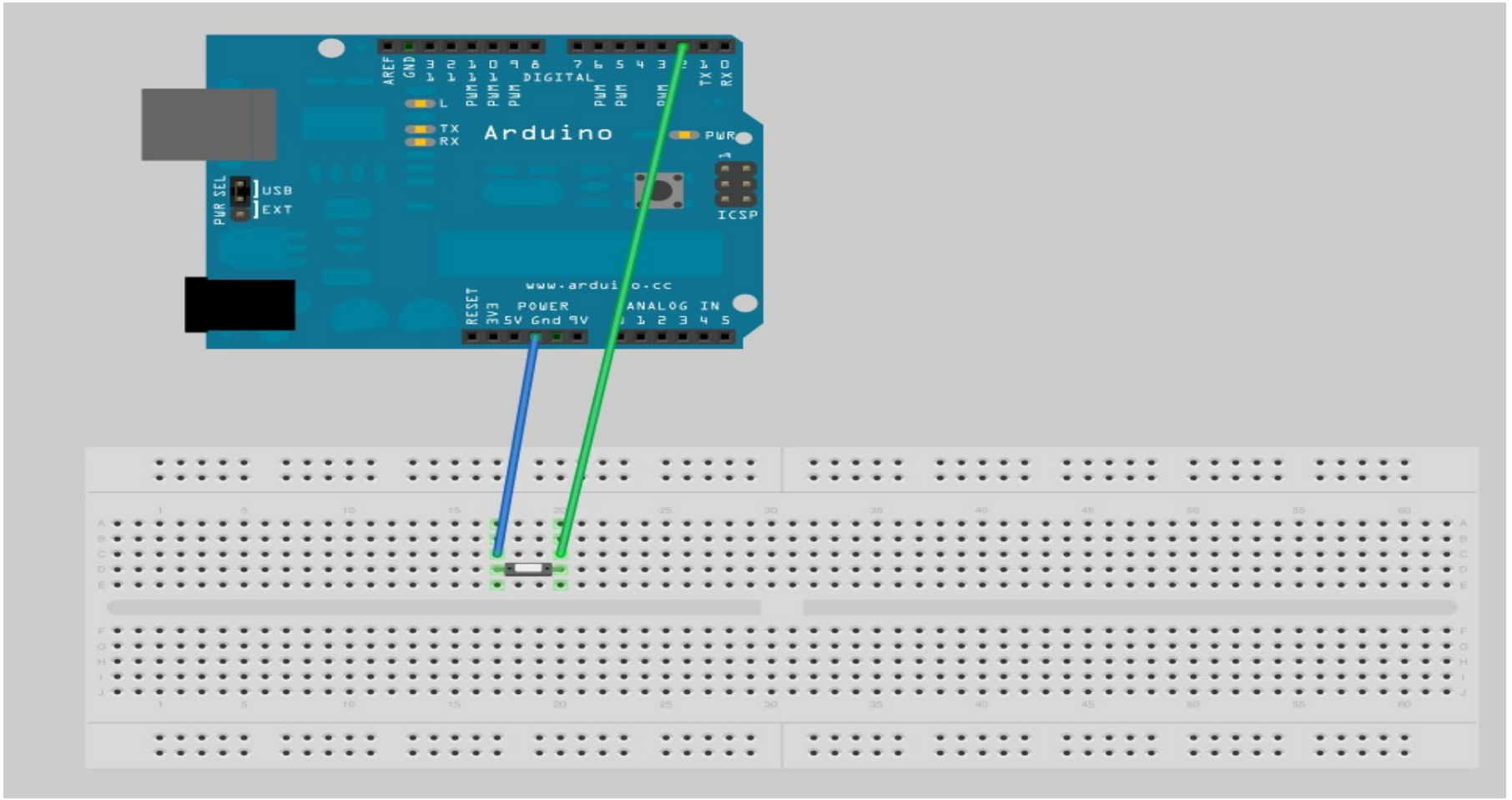


```
const int buttonPin = 2;     // the number of the pushbutton pin
const int ledPin =  13;      // the number of the LED pin

// variables will change:
int buttonState = 0;         // variable for reading the pushbutton status

void setup() {
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);
  digitalWrite(buttonPin,HIGH); //pullup resistors
}

void loop(){
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);

  // check if the pushbutton is pressed.
  // if it is, the buttonState is HIGH:
  if (buttonState == LOW) {     //pullup to high only to low if button pressed    //
    digitalWrite(ledPin, HIGH);
  }
  else {
    // turn LED off:
    digitalWrite(ledPin, LOW);
  }
}
```

your inbuilt resistor is now active

It`s called
PULL UP RESISTOR


pin 2
button
ground


AREF
GND
DIGITAL
L
TX
RX
Arduino
PWR
PWR SEL
USB
EXT
ICSP
www.arduino.cc
RESET
3V3
POWER
5V Gnd 9V
ANALOG IN
0 1 2 3 4 5

Now you are free to
replace your button with any
conductive material and connect it
temporarily through any thing that
comes to your mind

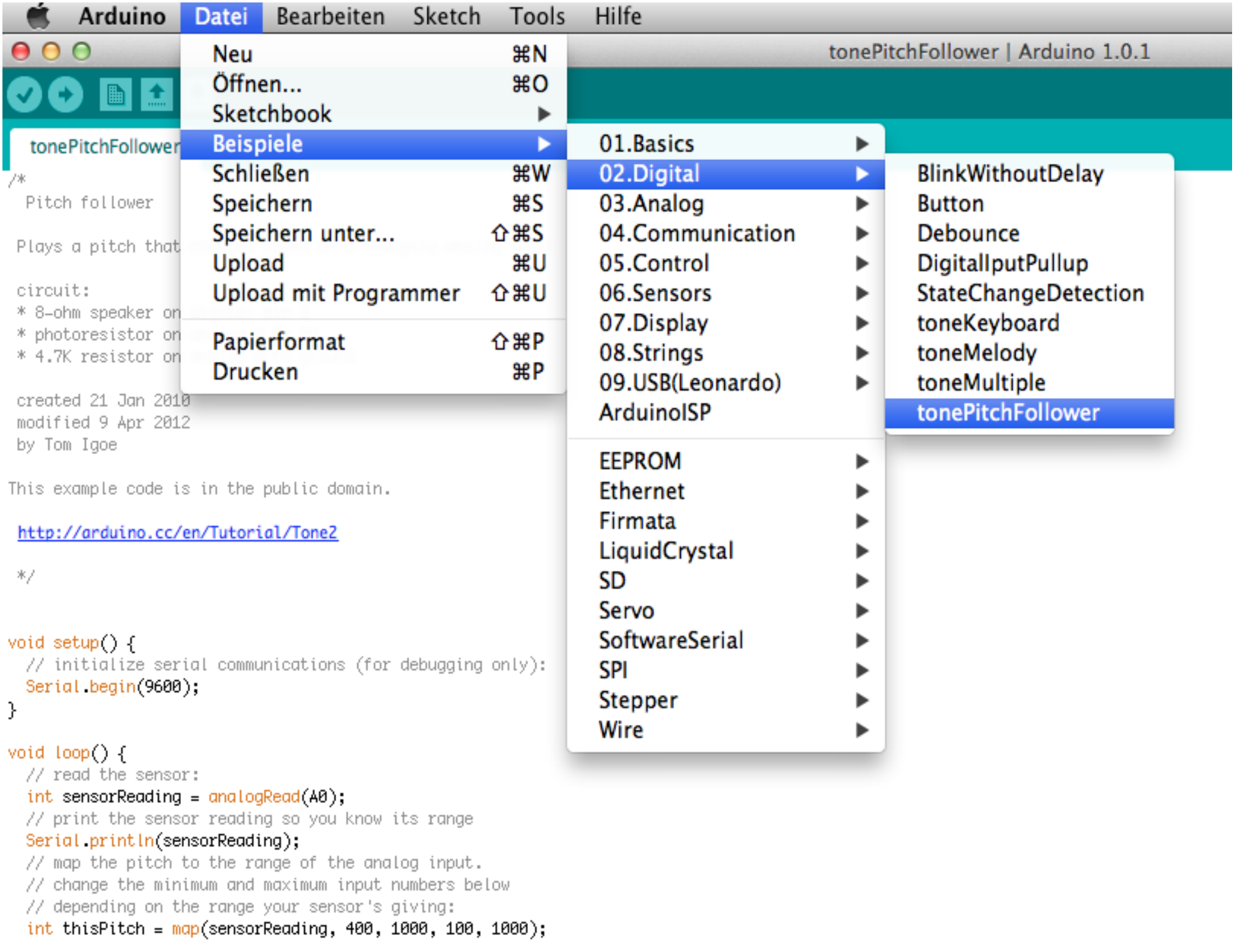
Leitender Faden

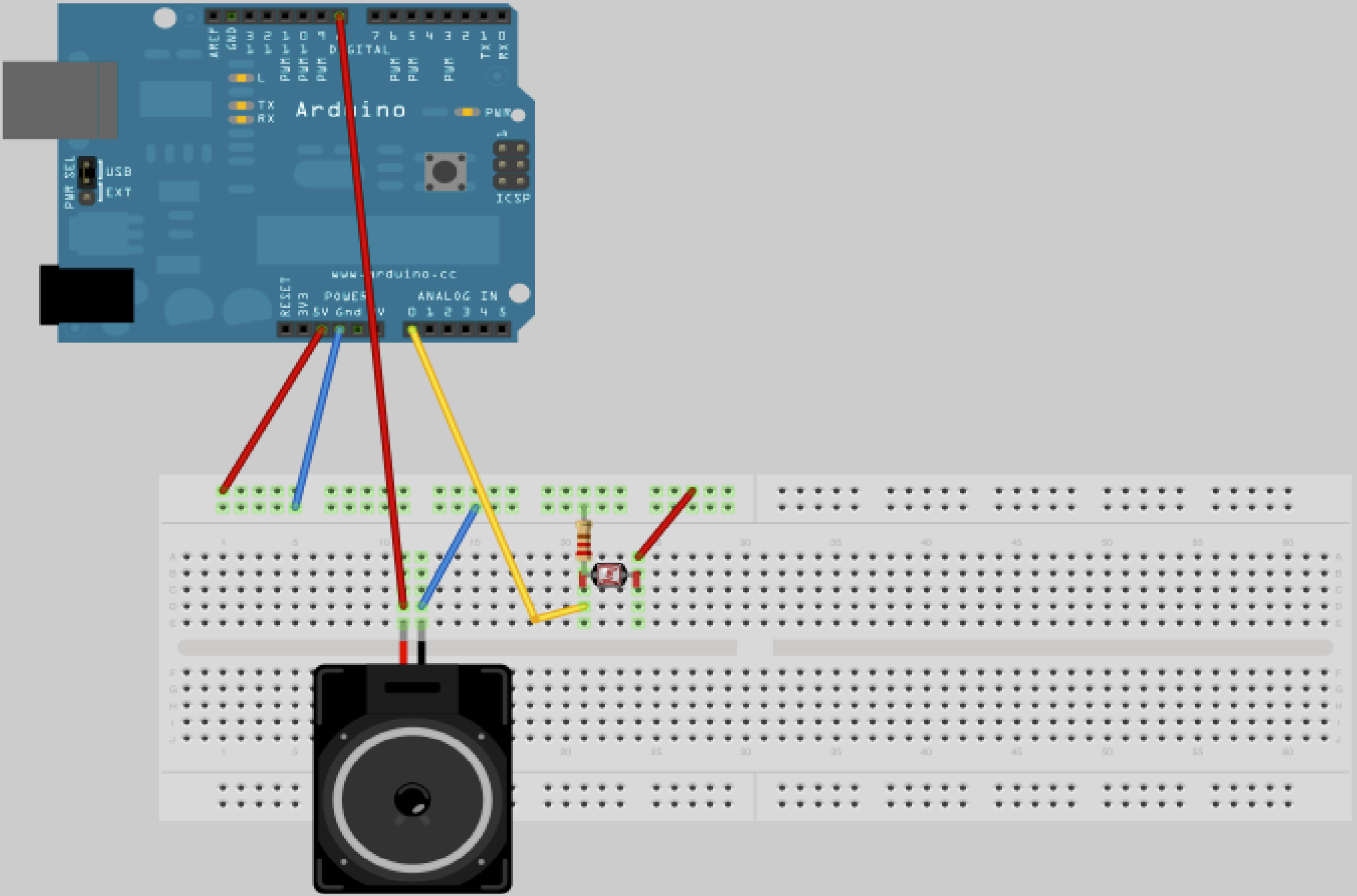
Leitender Stoff

Leitender Lack

Tone Pitch Follower

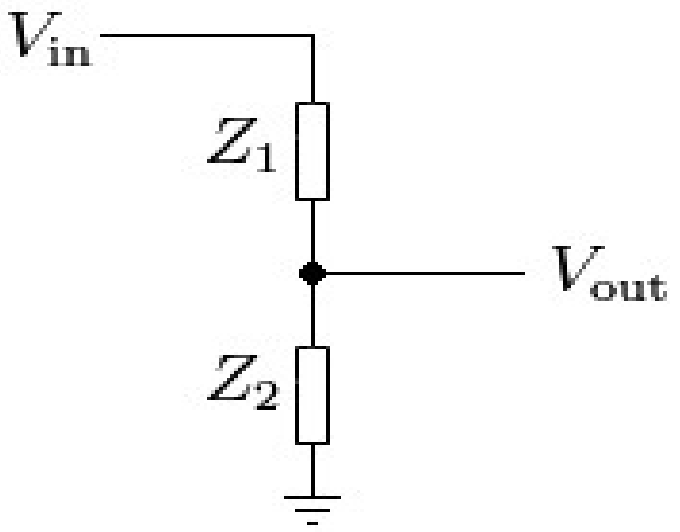
Examples – Digital - TonePitchFollower


Arduino
Datei
Bearbeiten
Sketch
Tools
Hilfe
tonePitchFollower | Arduino 1.0.1
Neu ⌘N
Öffnen... ⌘O
Sketchbook
Beispiele
Schließen ⌘W
Speichern ⌘S
Speichern unter... ⇧⌘S
Upload ⌘U
Upload mit Programmer ⇧⌘U
Papierformat ⇧⌘P
Drucken ⌘P
01.Basics
02.Digital
03.Analog
04.Communication
05.Control
06.Sensors
07.Display
08.Strings
09.USB(Leonardo)
ArduinoISP
EEPROM
Ethernet
Firmata
LiquidCrystal
SD
Servo
SoftwareSerial
SPI
Stepper
Wire
BlinkWithoutDelay
Button
Debounce
DigitalInputPullup
StateChangeDetection
toneKeyboard
toneMelody
toneMultiple
tonePitchFollower
tonePitchFollower
/*
Pitch follower
Plays a pitch that
circuit:
* 8-ohm speaker on
* photoresistor on
* 4.7K resistor on
created 21 Jan 2010
modified 9 Apr 2012
by Tom Igoe
This example code is in the public domain.
http://arduino.cc/en/Tutorial/Tone2
*/
void setup() {
// initialize serial communications (for debugging only):
Serial.begin(9600);
}
void loop() {
// read the sensor:
int sensorReading = analogRead(A0);
// print the sensor reading so you know its range
Serial.println(sensorReading);
// map the pitch to the range of the analog input.
// change the minimum and maximum input numbers below
// depending on the range your sensor's giving:
int thisPitch = map(sensorReading, 400, 1000, 100, 1000);


AREF
GND
DIGITAL
PWM
TX
RX
L
Arduino
PWR
PWR SEL
USB
EXT
ICSP
www.arduino.cc
RESET
POWER
ANALOG IN
5V
Gnd
0 1 2 3 4 5

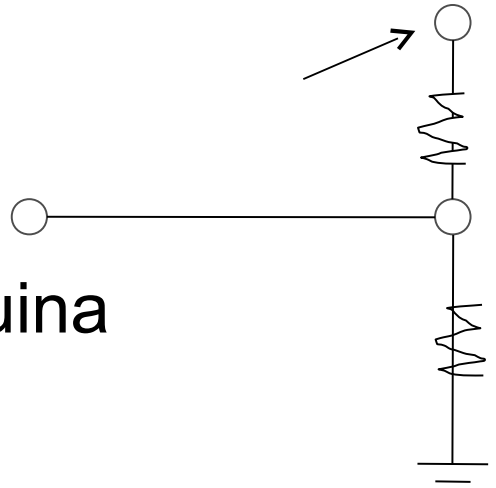
spannungsteiler

$$V_{\text{out}} = \frac{Z_2}{Z_1 + Z_2} \cdot V_{\text{in}}$$



Self made potentiometer

Voltage devider

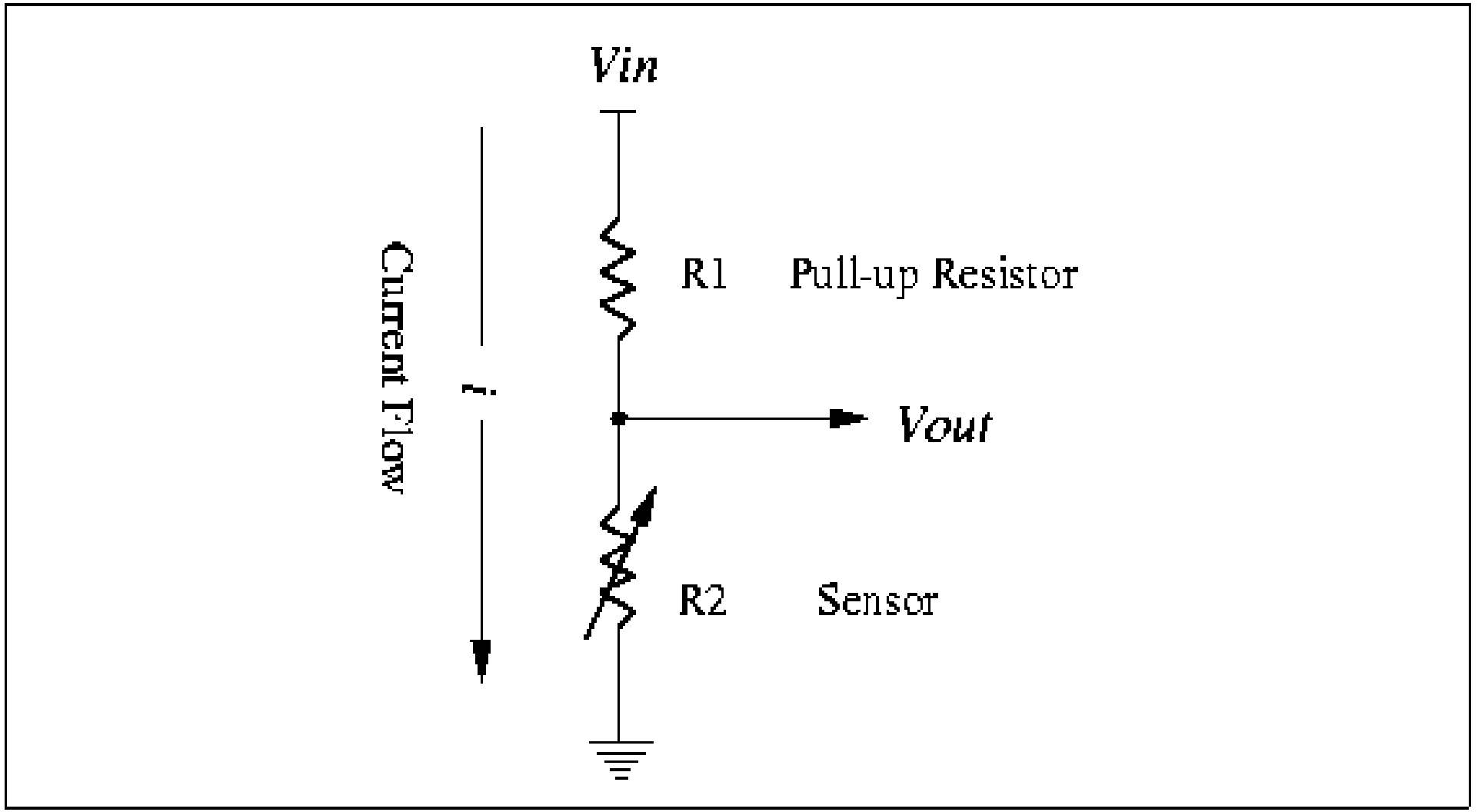


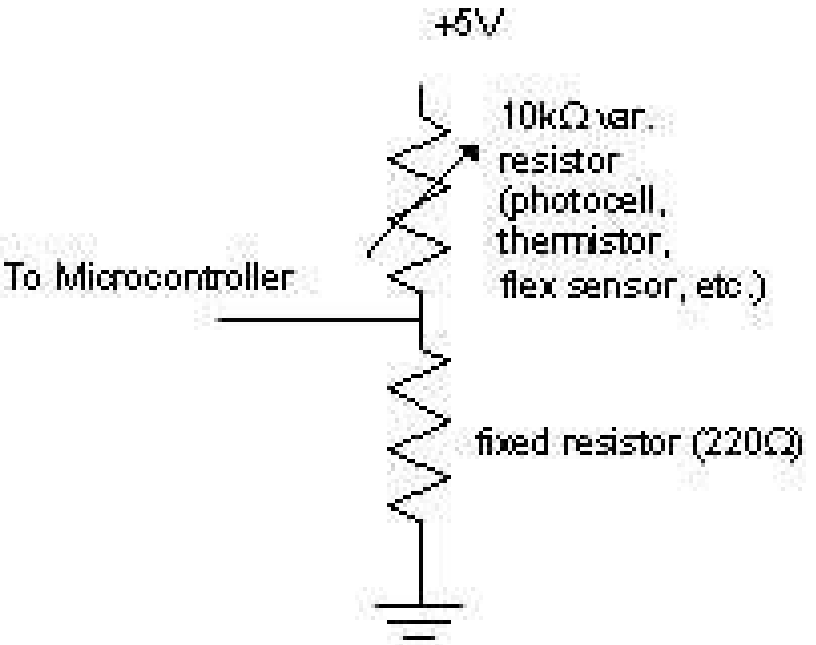
PIN to arduina

ground

Variable resistance
from our sensor

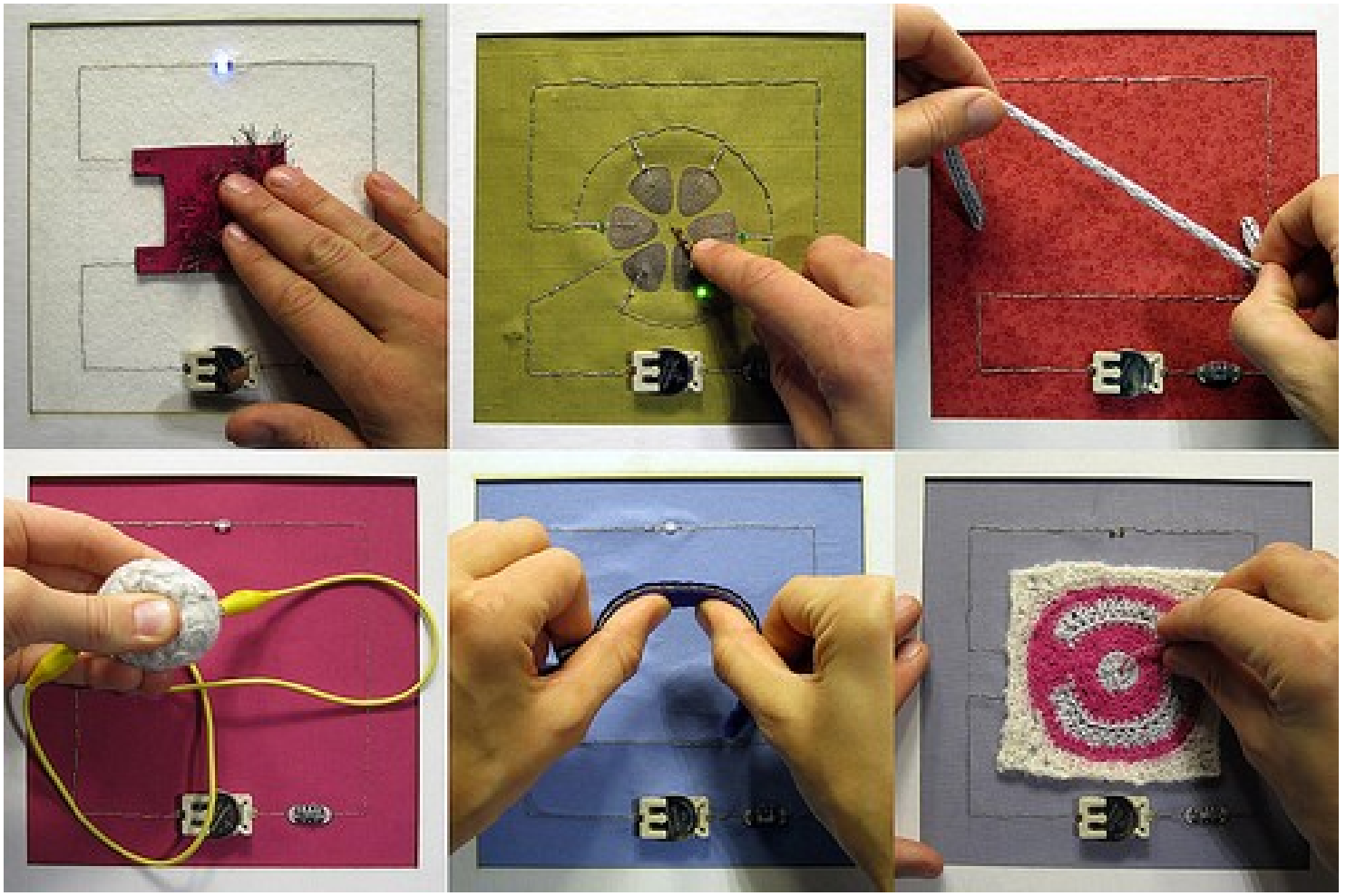
Stable resistance
as big as the maximum
resistance
(best you take 2k Ohm)


Vin
R1
Pull-up Resistor
Vout
Current Flow
i
R2
Sensor


+5V
10kΩ var. resistor (photocell, thermistor, flex sensor, etc.)
To Microcontroller
fixed resistor (220Ω)

mz baltazar's laboratory

hannah perner wilson

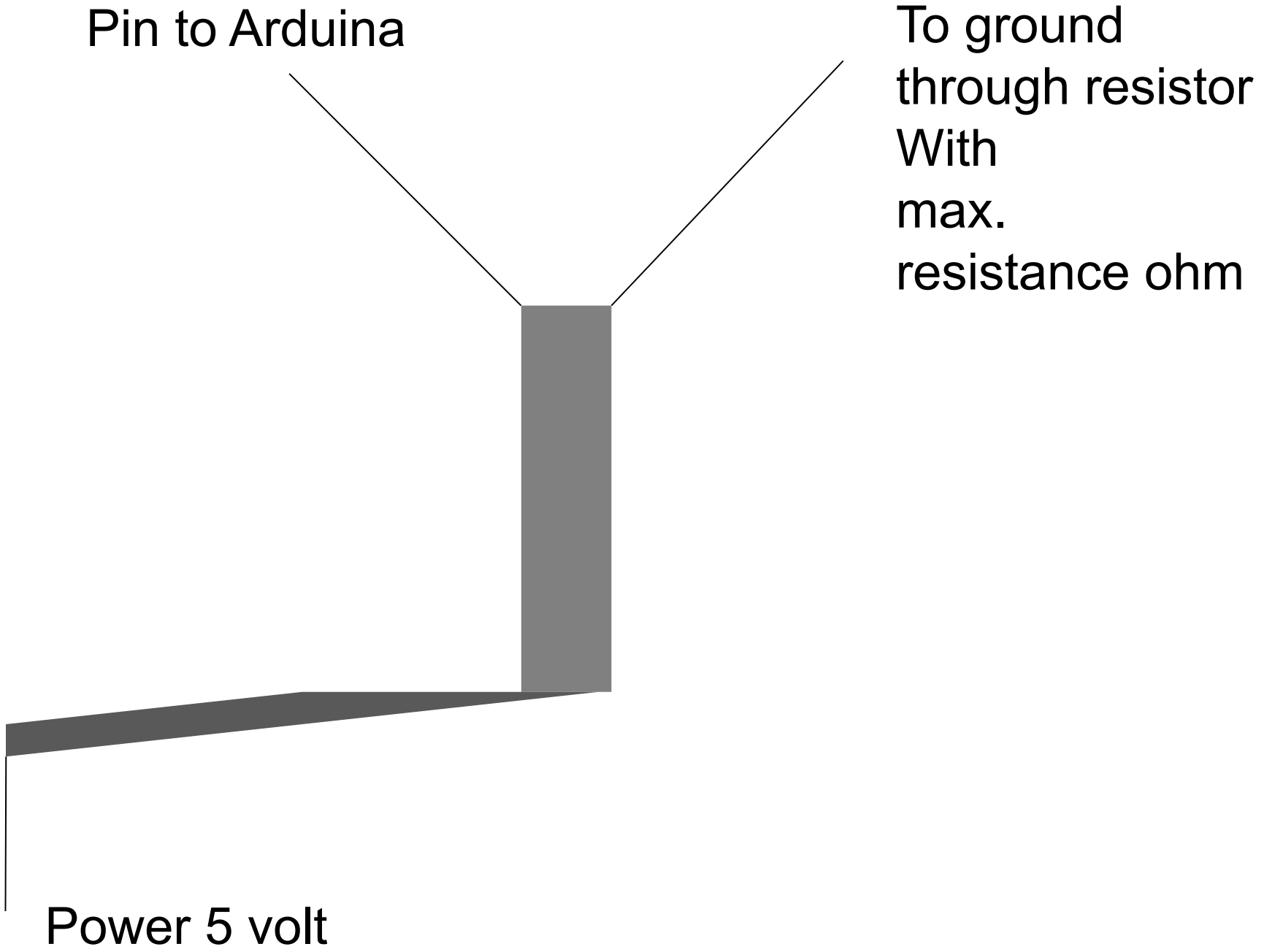


3 LEGS:

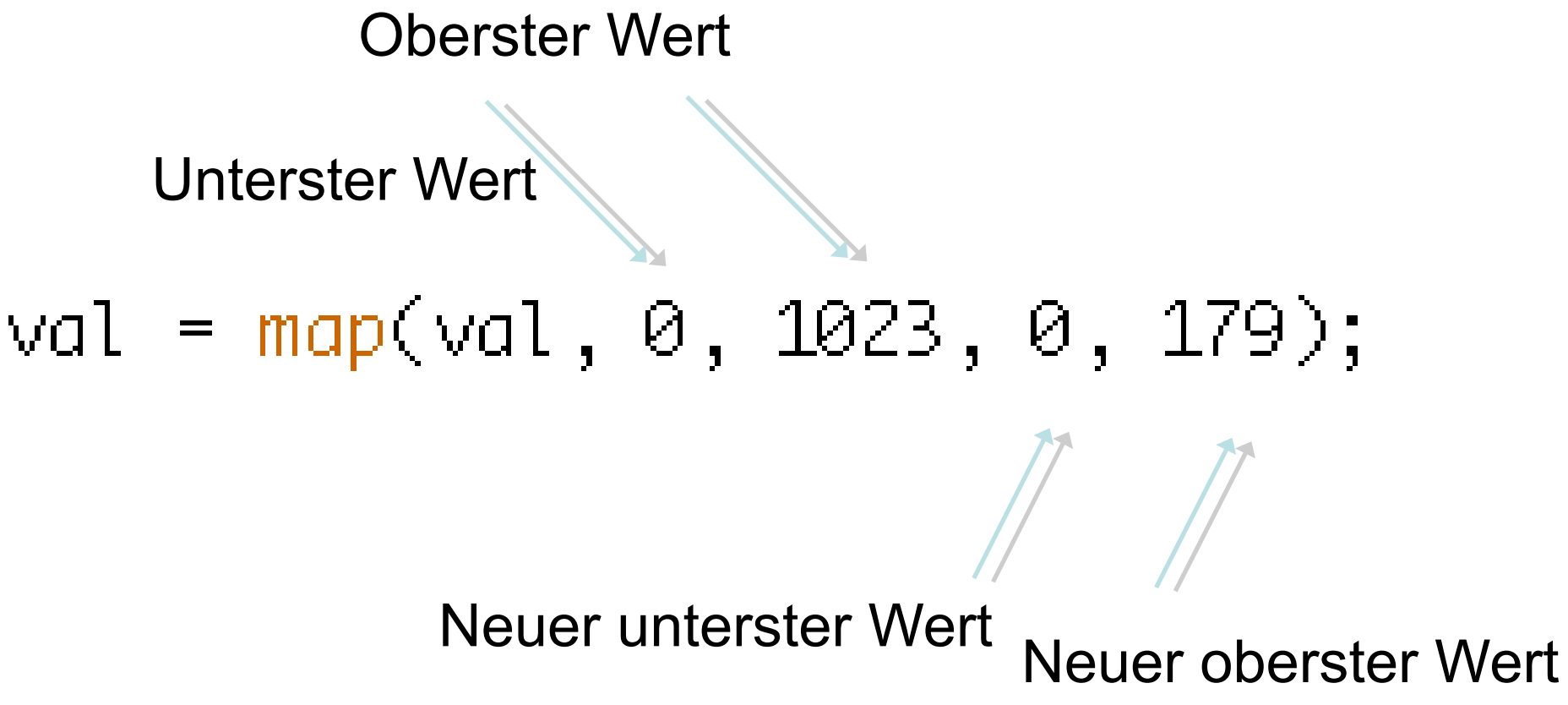
One to power

One through the resistor to ground

One to Arduino Pin „A0“


Pin to Arduina
To ground
through resistor
With
max.
resistance ohm
Power 5 volt

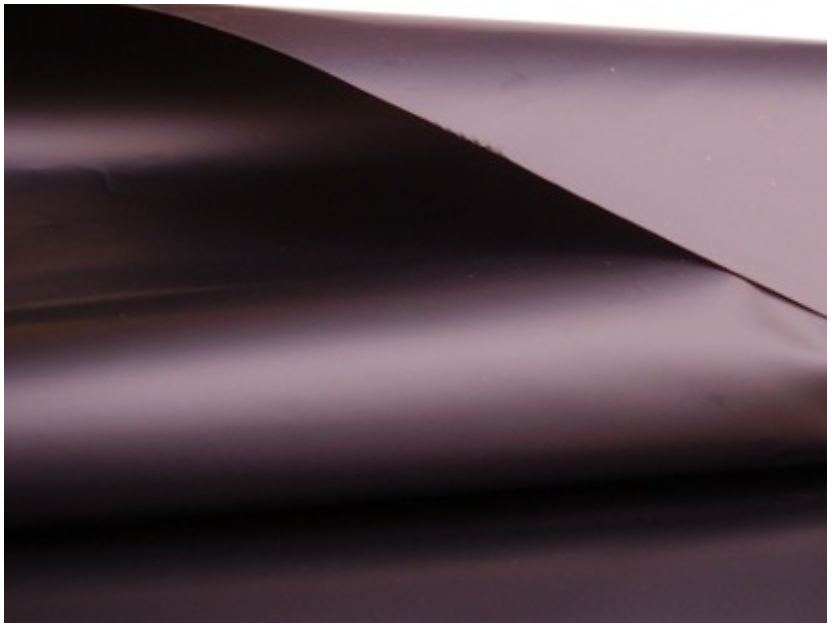
Map


Oberster Wert
Unterster Wert
val = map(val, 0, 1023, 0, 179);
Neuer unterster Wert
Neuer oberster Wert

RESISTIVE MATERIAL

VELOSTAT

http://www.plugandwear.com/default.asp?mod=product&cat_id=89,104&product_id=136



http://www.kobakant.at/DIY/?p=381

Leitender Faden

http://www.plugandwear.com/default.asp?mod=product&cat_id=105&product_id=137

Leitender Stoff

http://www.plugandwear.com/default.asp?mod=product&cat_id=89,104&product_id=138

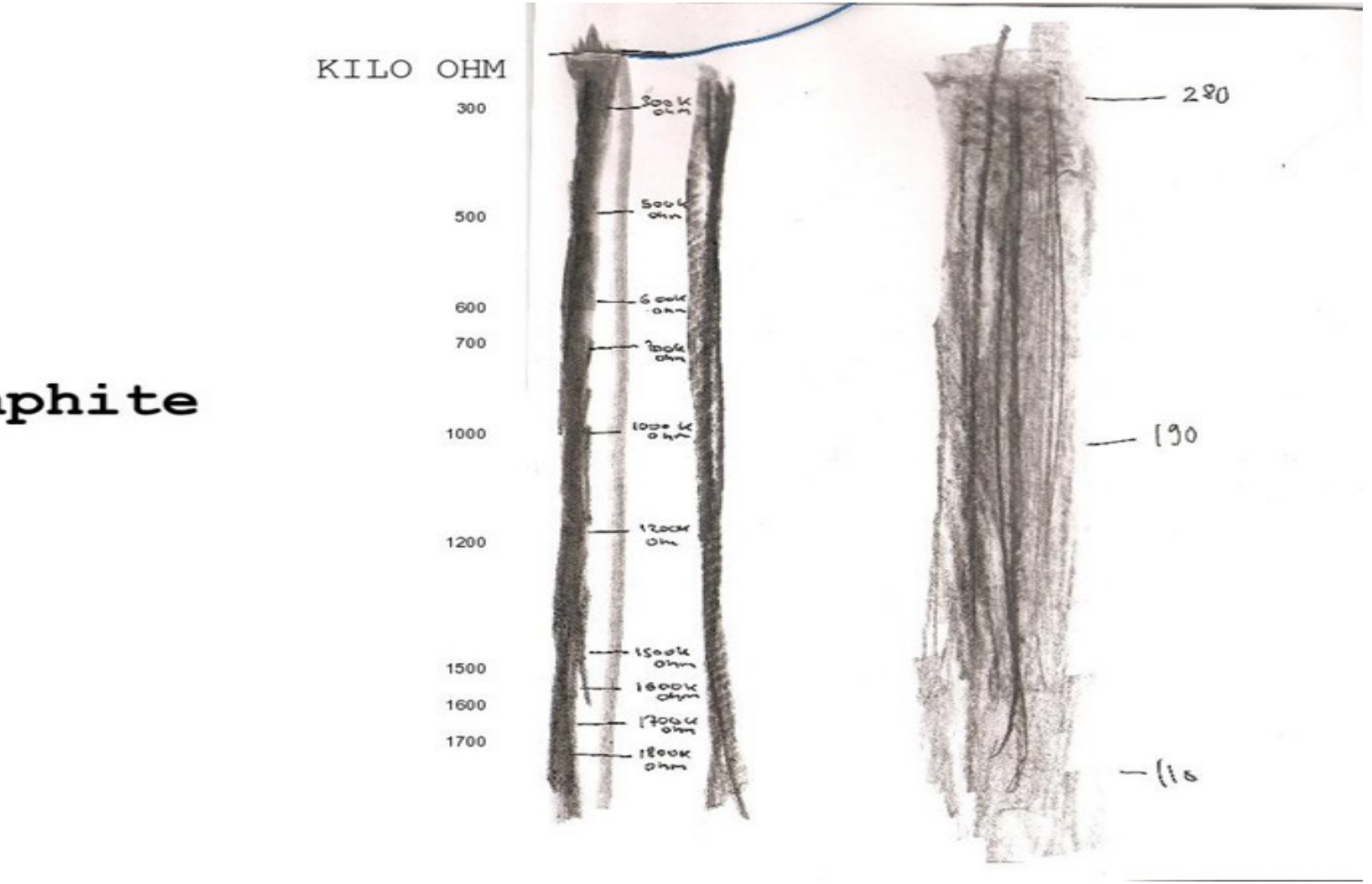
www.physicalcomputing.at

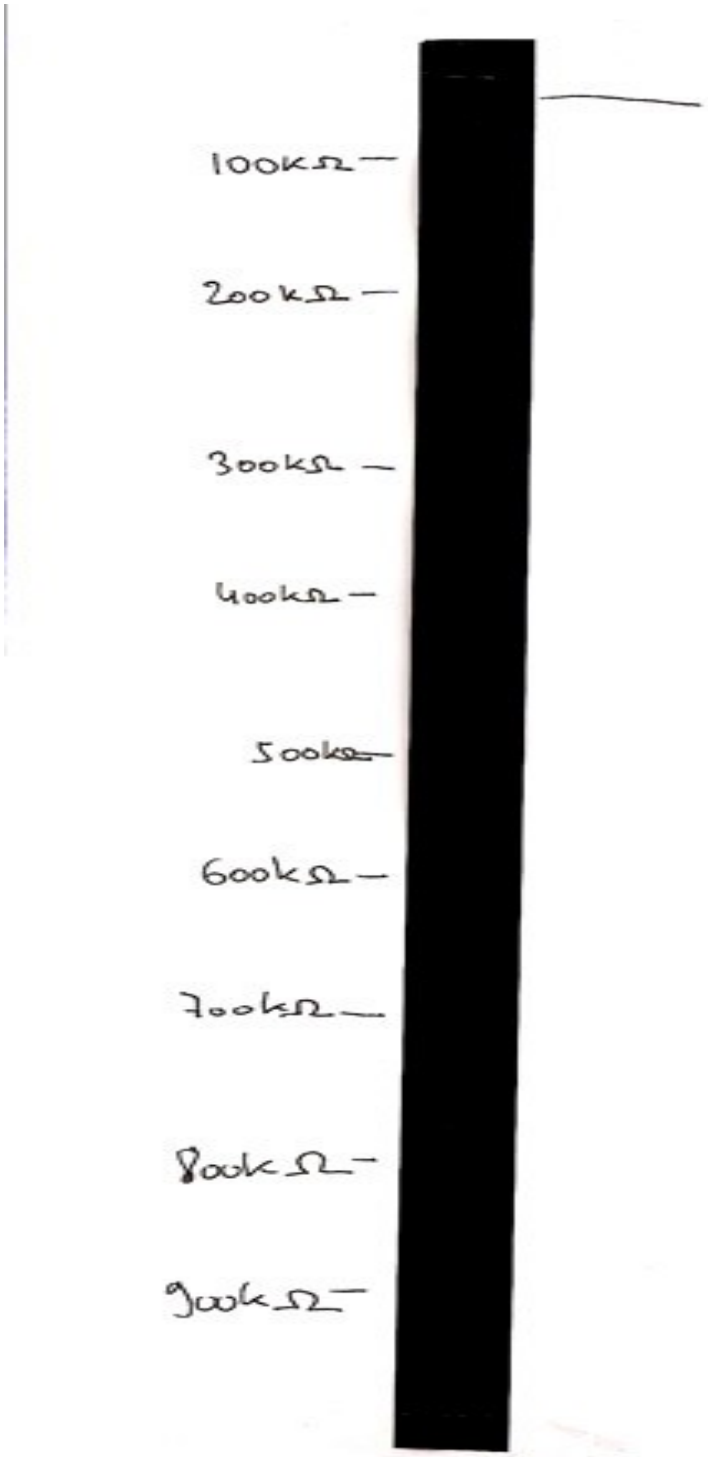
possible materials to use as potentiometers

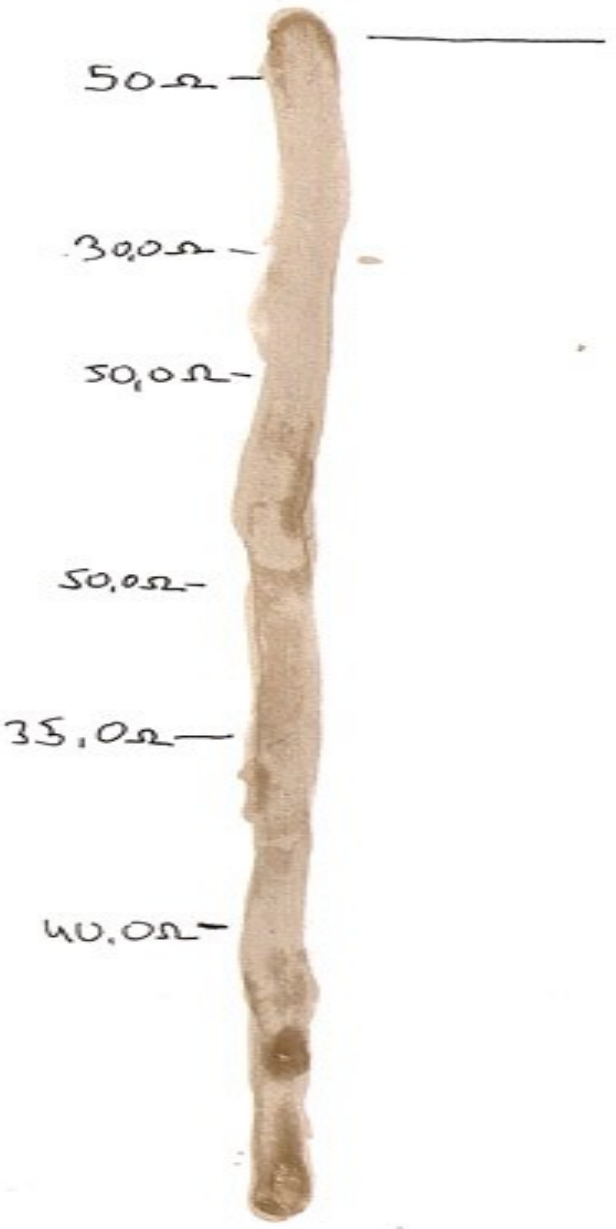
Videotape

Graphite

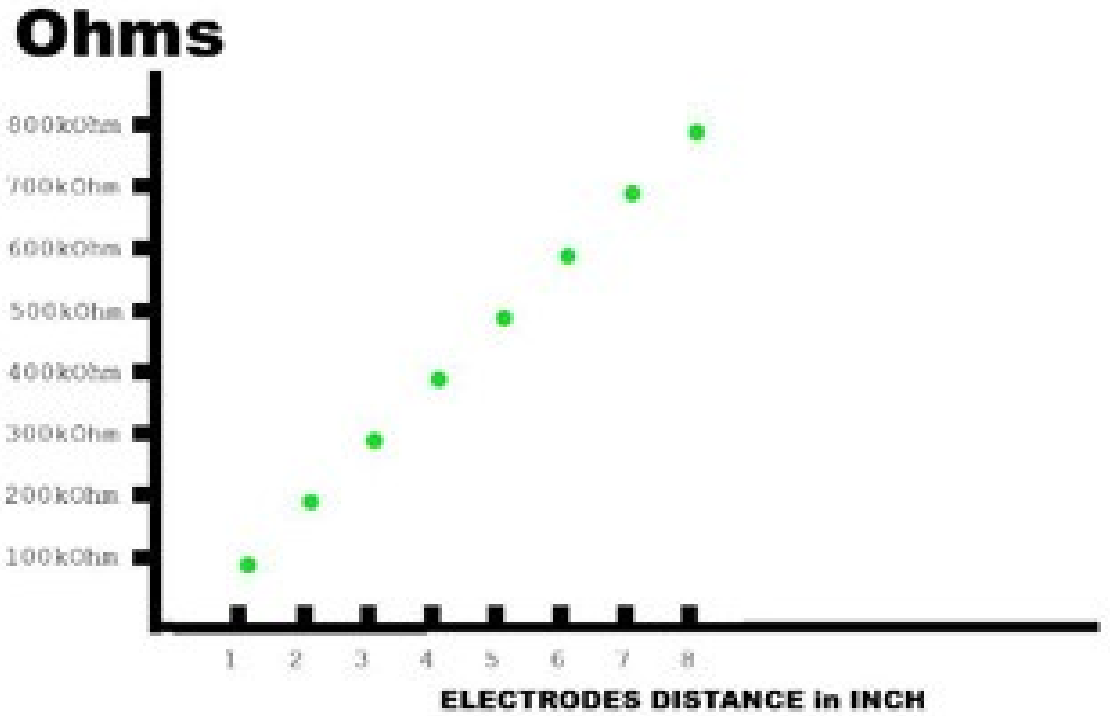
Graphite




100kΩ
200kΩ
300kΩ
400kΩ
500kΩ
600kΩ
700kΩ
800kΩ
900kΩ

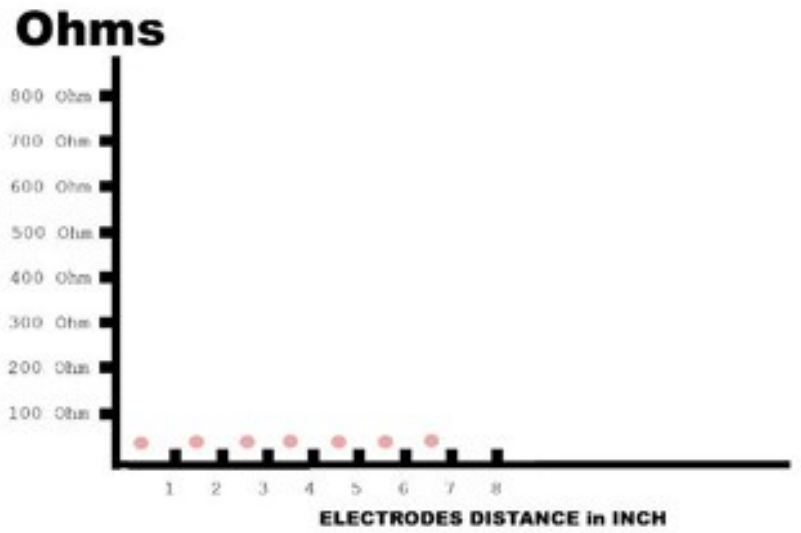

50Ω
30,0Ω
50,0Ω
50,0Ω
35,0Ω
40,0Ω

Only valid for video tape produced before 1996

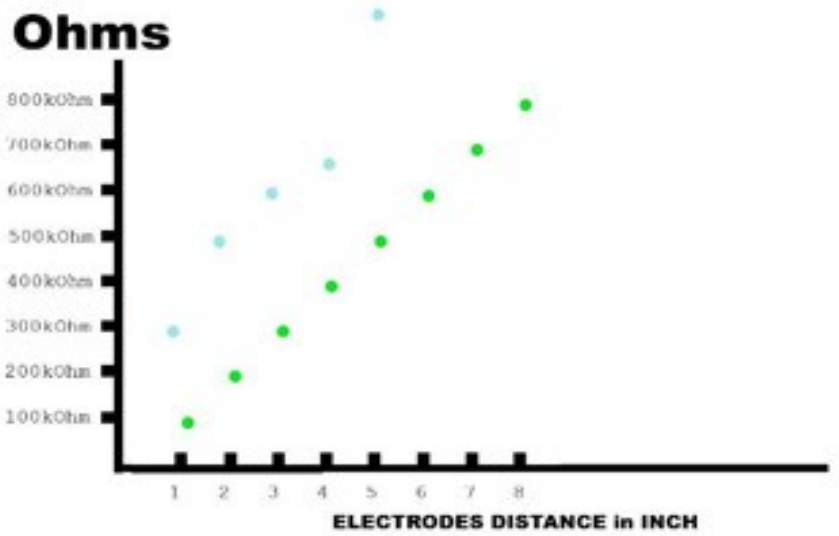


Conductive Ink 30 - 50 Ohms

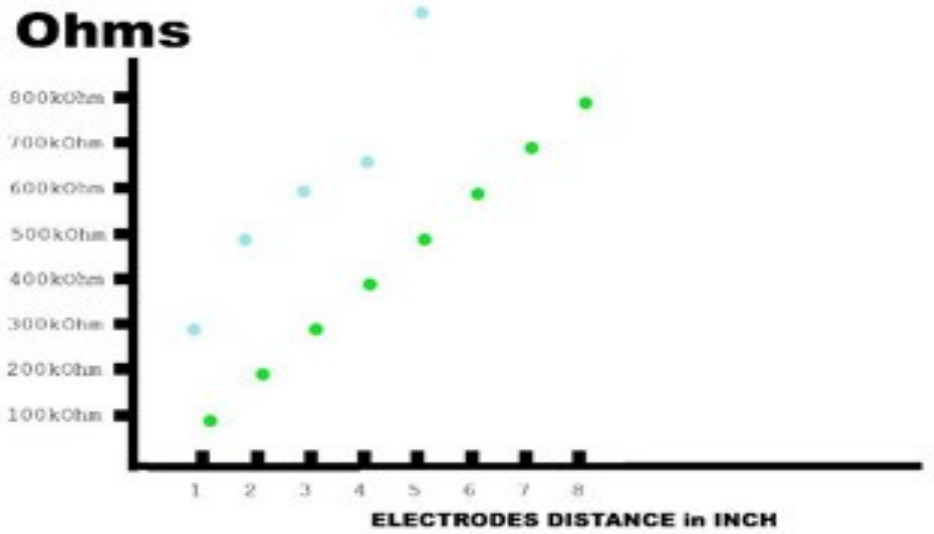
Disadvantage: no uniformed conductivity



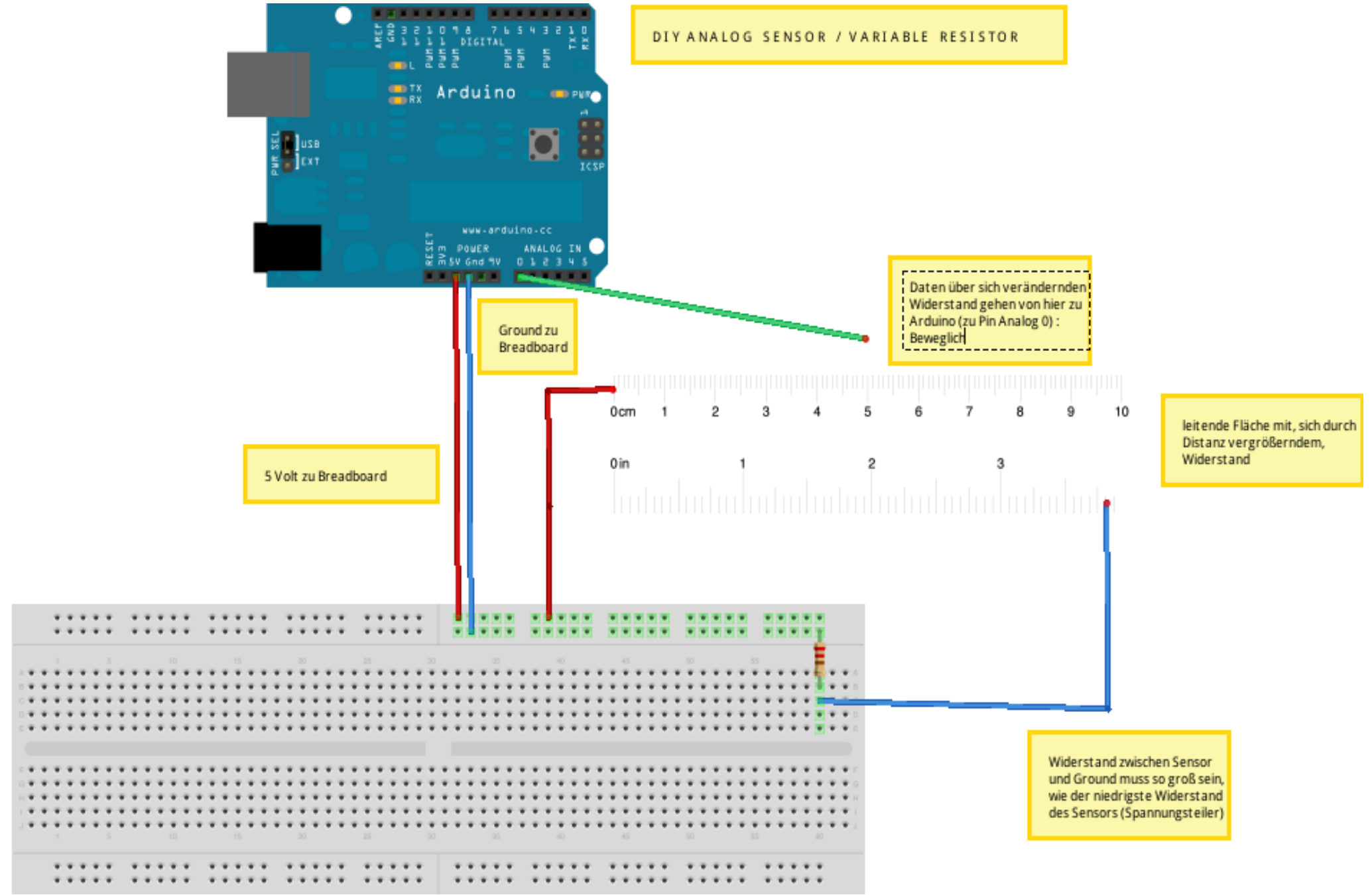
Graphite vs. Videotape

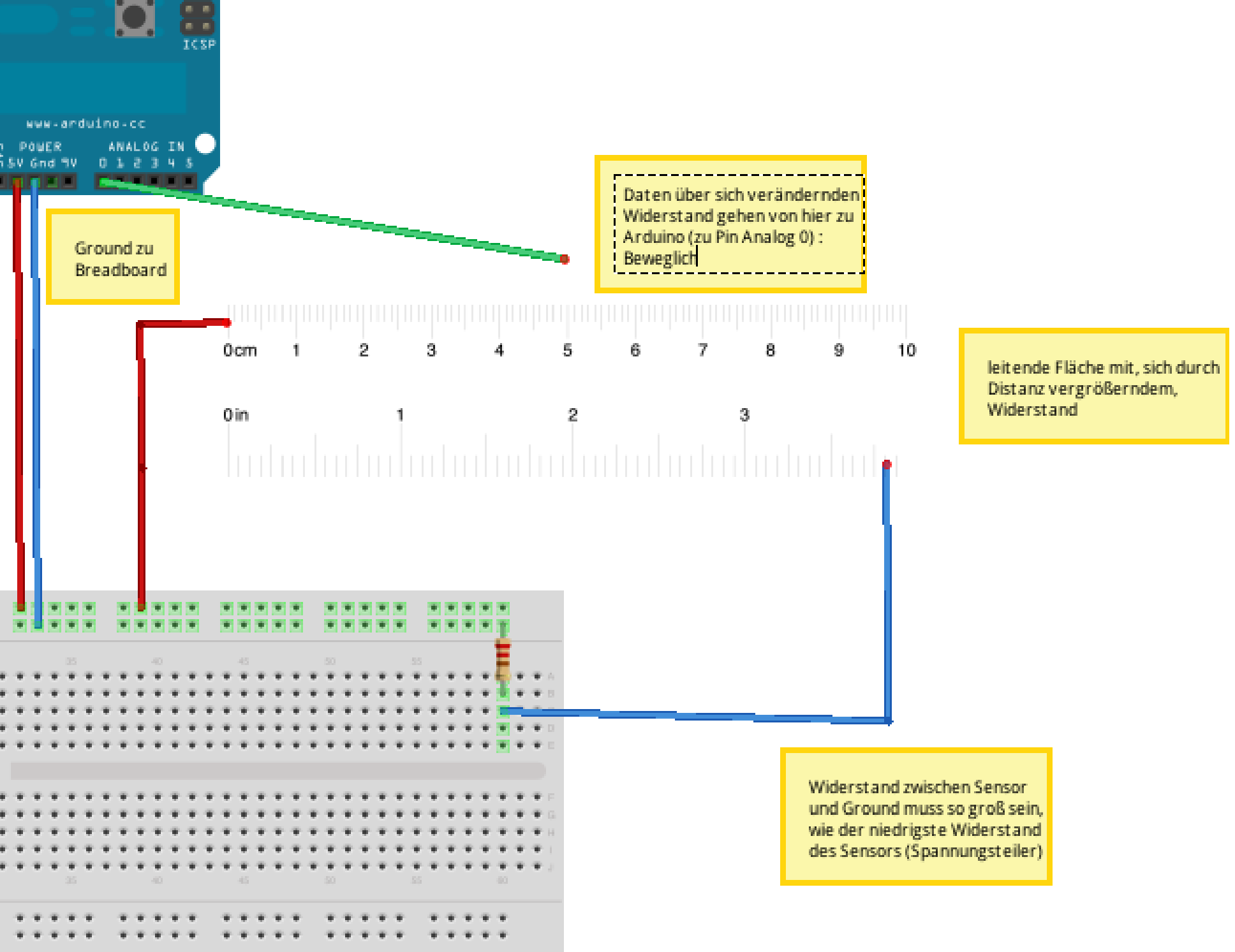


Graphite vs. Videotape

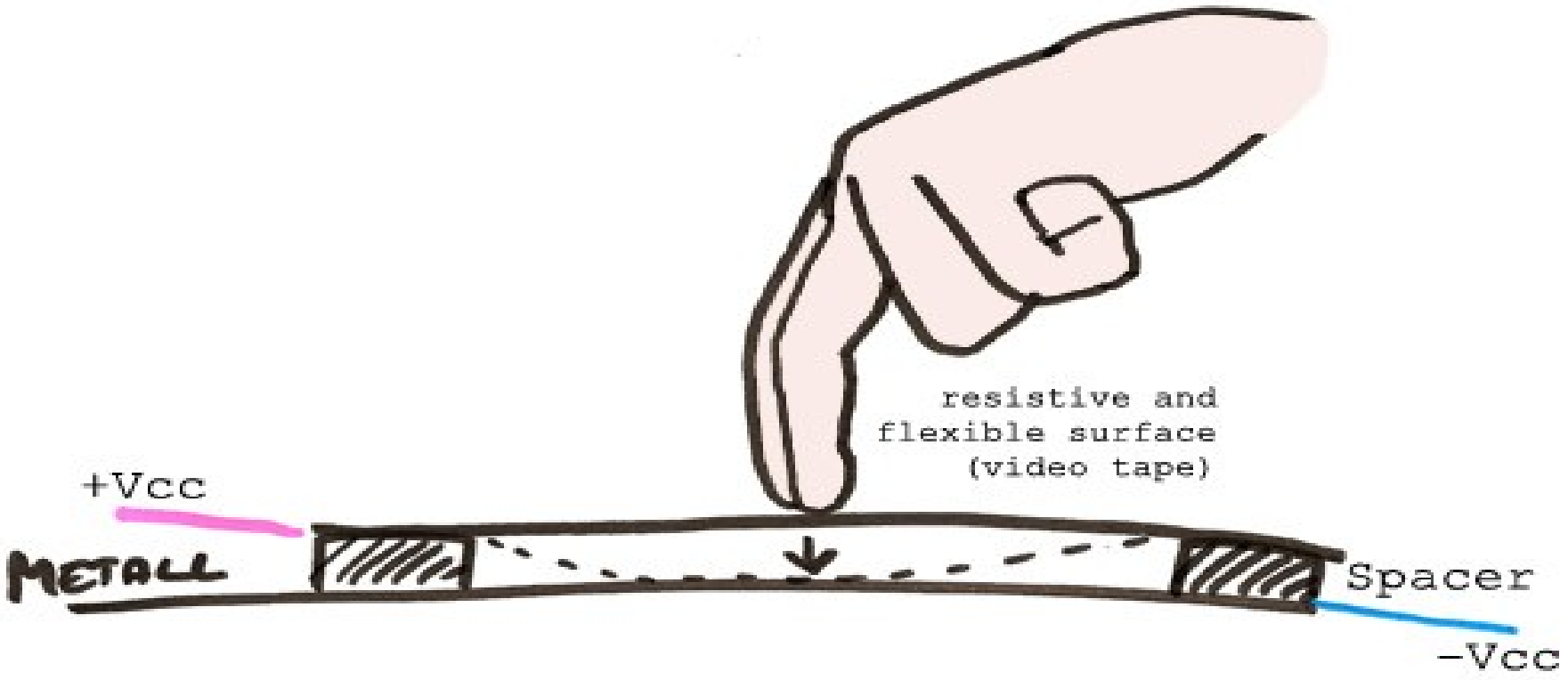


graphit


DIY ANALOG SENSOR / VARIABLE RESISTOR
Arduino
www.arduino.cc
Daten über sich verändernden Widerstand gehen von hier zu Arduino (zu Pin Analog 0) : Beweglich
Ground zu Breadboard
5 Volt zu Breadboard
leitende Fläche mit, sich durch Distanz vergrößerndem, Widerstand
Widerstand zwischen Sensor und Ground muss so groß sein, wie der niedrigste Widerstand des Sensors (Spannungsteiler)

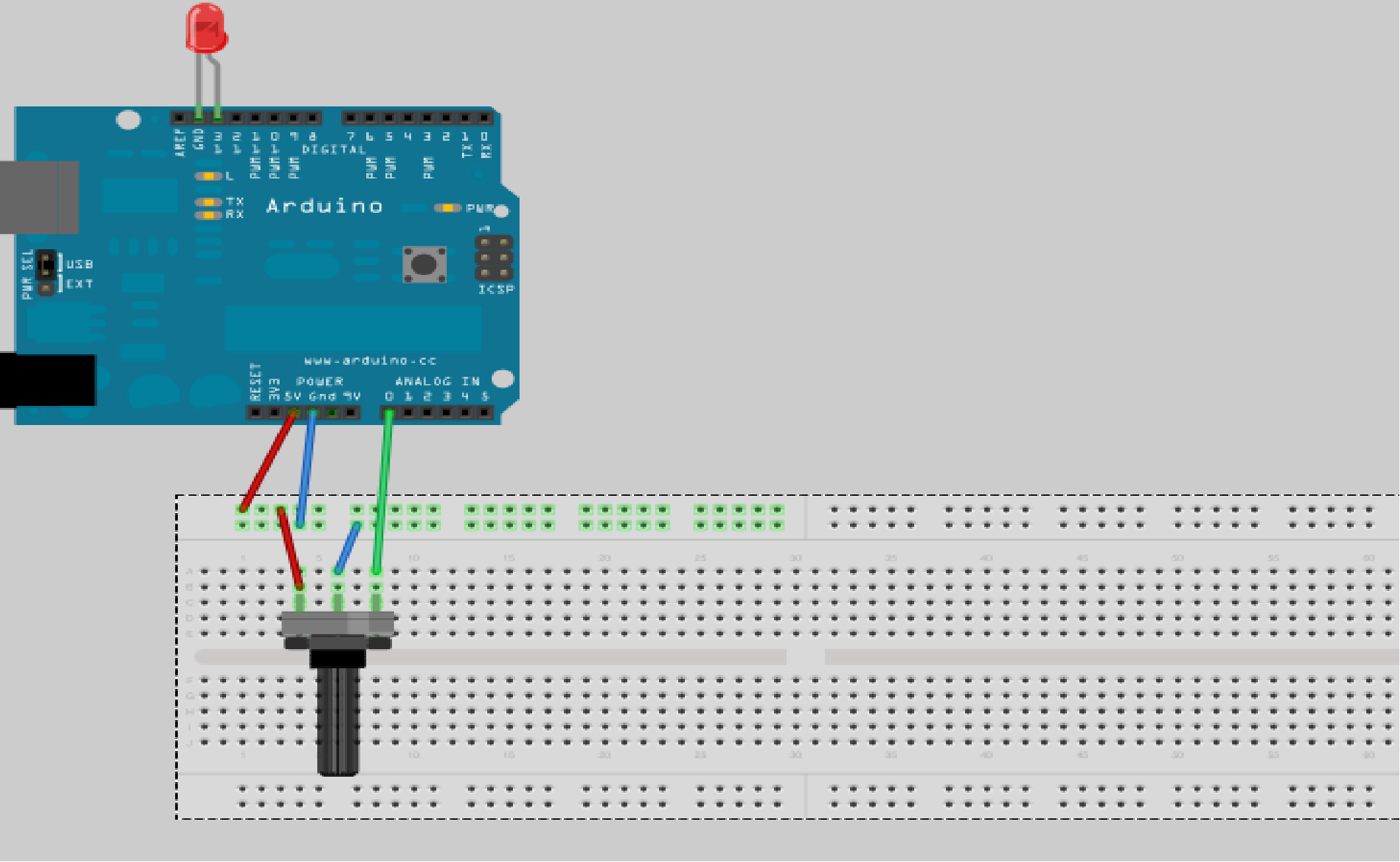

ICSP
www.arduino.cc
POWER
ANALOG IN
0 1 2 3 4 5
Ground zu Breadboard
Daten über sich verändernden Widerstand gehen von hier zu Arduino (zu Pin Analog 0) : Beweglich
0cm 1 2 3 4 5 6 7 8 9 10
0in 1 2 3
leitende Fläche mit, sich durch Distanz vergrößerndem, Widerstand
Widerstand zwischen Sensor und Ground muss so groß sein, wie der niedrigste Widerstand des Sensors (Spannungsteiler)

Touch sensor



MESSGERÄT FÜR SELBST GEMACHTE SENSOREN

mz baltazar's laboratory

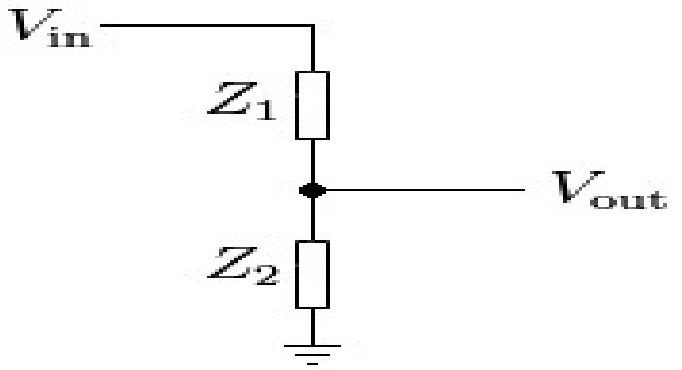

AREF
GND
DIGITAL
PWM
TX
RX
L
Arduino
PWR
USB
EXT
PWR SEL
ICSP
www.arduino.cc
RESET
3V3
5V
Gnd
Vin
POWER
ANALOG IN
0 1 2 3 4 5

Arduino Code

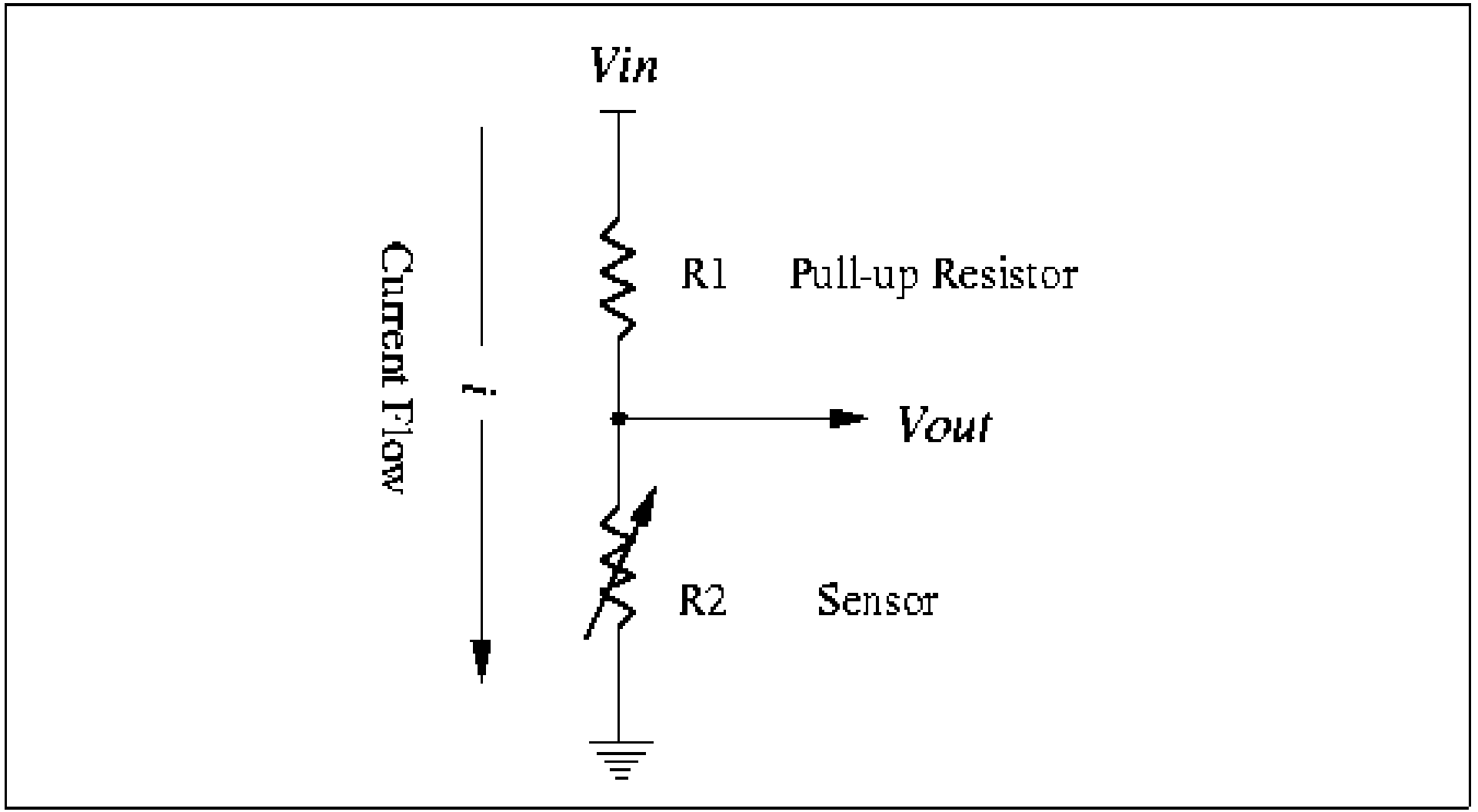
Examples – Analog - AnalogInOutSerial

```
Serial

nstants won't change.  They're used to give names
ins used:
nalogInPin = A0;  // Analog input pin that the potentiometer is attached to
nalogOutPin = 9; // Analog output pin that the LED is attached to

alue = 0;         // value read from the pot
alue = 0;         // value output to the PWM (analog out)

) {
lize serial communications at 9600 bps:
gin(9600);


 {
he analog in value:
ue = analogRead(analogInPin);
 to the range of the analog out:
ue = map(sensorValue, 0, 1023, 0, 255);
 the analog out value:
te(analogOutPin, outputValue);

the results to the serial monitor:
int("sensor = " );
int(sensorValue);
int("\t output = ");
intln(outputValue);

0 milliseconds before the next loop
e analog-to-digital converter to settle
the last reading:
;
```

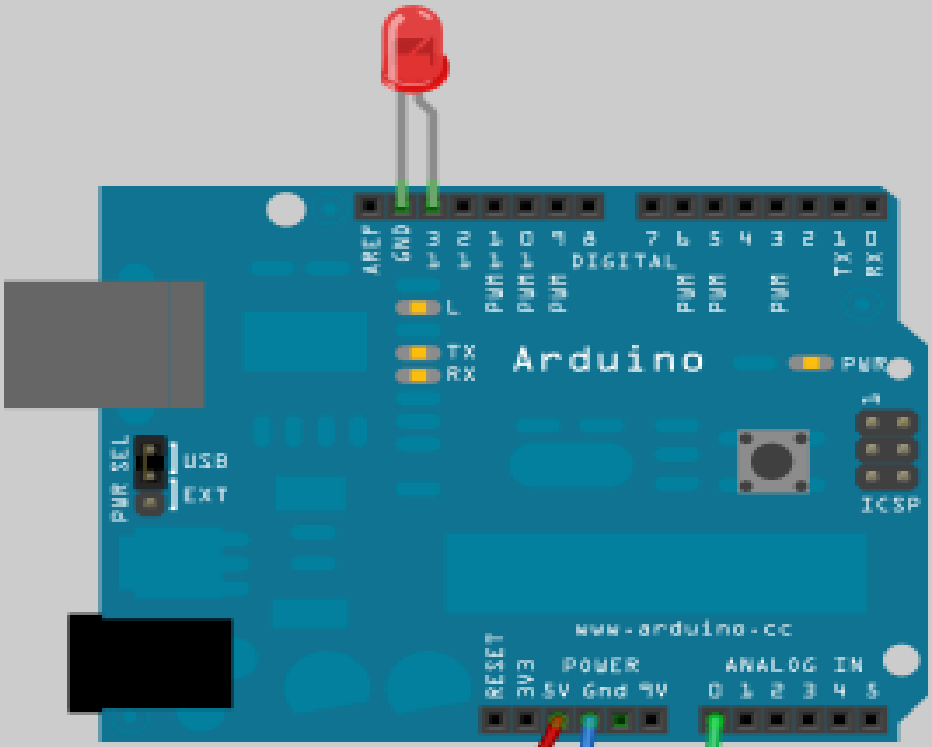
spannungsteiler



$$V_{out} = \frac{Z_2}{Z_1 + Z_2} \cdot V_{in}$$


Vin
R1
Pull-up Resistor
Vout
Current Flow
i
R2
Sensor

FOR LOOP
WHILE LOOP


AREF
GND
DIGITAL
L
TX
RX
Arduino
PWR
PWR SEL
USB
EXT
ICSP
www.arduino.cc
RESET
3V3
POWER
5V
Gnd
9V
ANALOG IN
0 1 2 3 4 5

```
outputValue = map(sensorValue, 0, 1023, 0, 255);
```

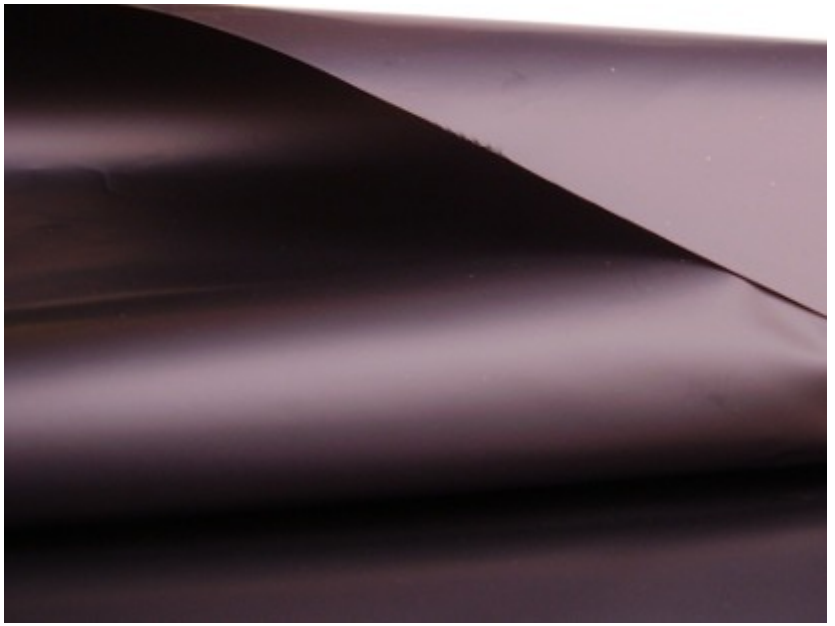
Leitender Faden

Leitender Stoff

Leitender Lack

VELOSTAT

http://www.plugandwear.com/default.asp?mod=product&cat_id=89,104&product_id=136



Leitender Faden

http://www.plugandwear.com/default.asp?mod=product&cat_id=105&product_id=137

Leitender Stoff

http://www.plugandwear.com/default.asp?mod=product&cat_id=89,104&product_id=138

www.physicalcomputing.at

possible materials to use as potentiometers

Videotape

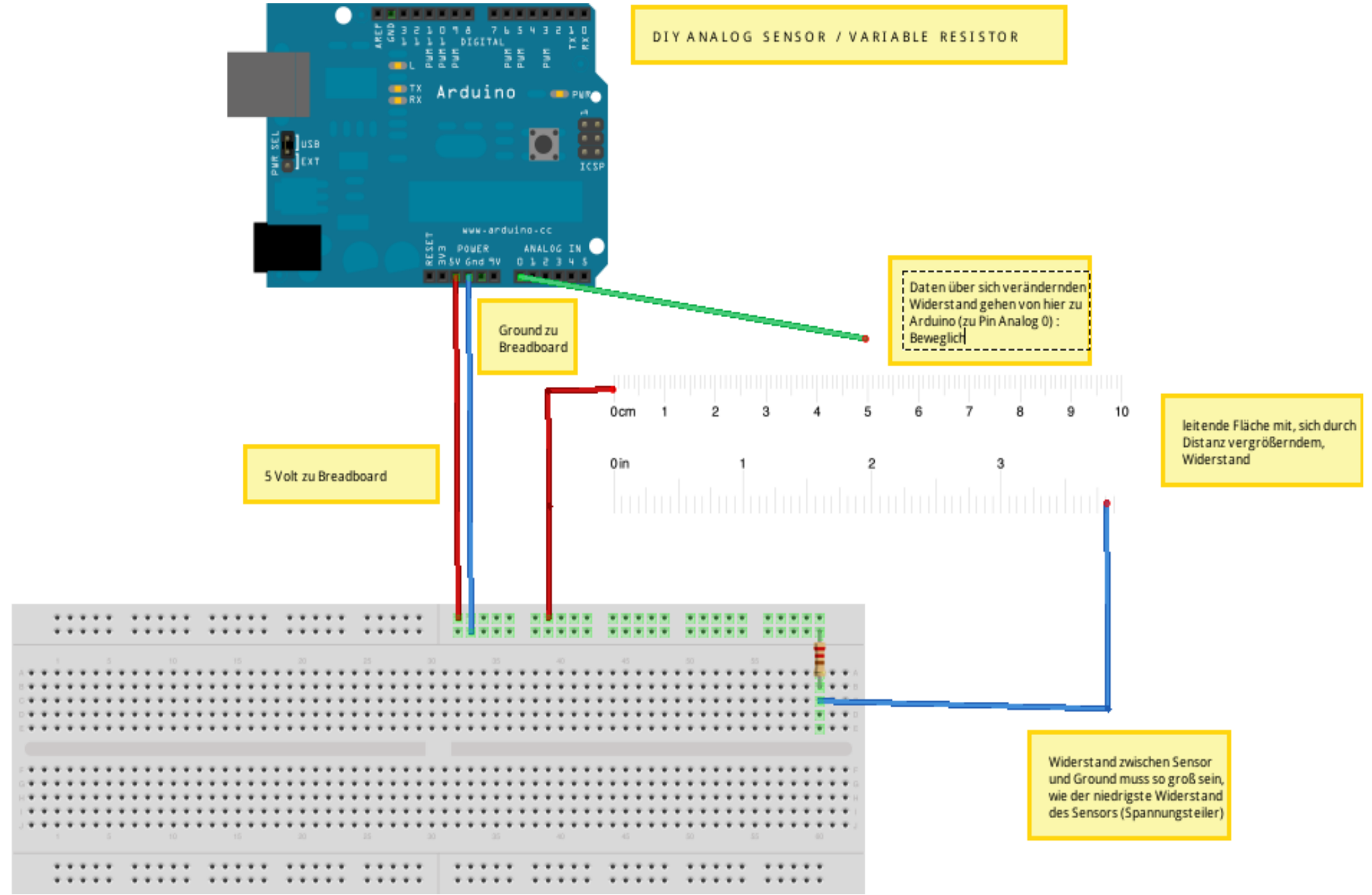
Graphite

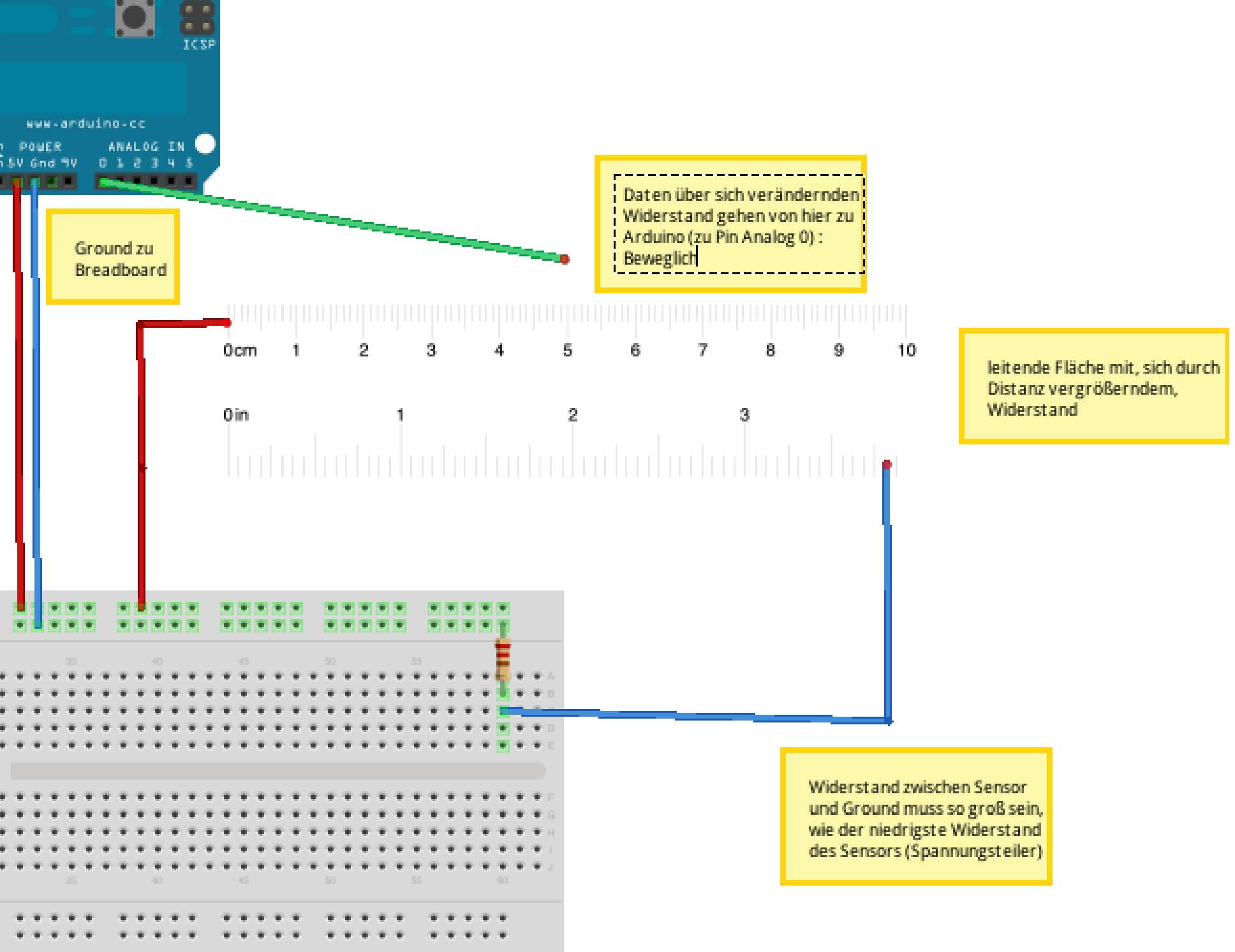
Pressure Sensor

Plastik (Isolating material)

Velostat

Plastik (Isolating material)


DIY ANALOG SENSOR / VARIABLE RESISTOR
Arduino
www.arduino.cc
Daten über sich verändernden Widerstand gehen von hier zu Arduino (zu Pin Analog 0) : Beweglich
Ground zu Breadboard
5 Volt zu Breadboard
leitende Fläche mit, sich durch Distanz vergrößerndem, Widerstand
Widerstand zwischen Sensor und Ground muss so groß sein, wie der niedrigste Widerstand des Sensors (Spannungsteiler)


ICSP
www.arduino.cc
POWER
ANALOG IN
0 1 2 3 4 5
Ground zu Breadboard
Daten über sich verändernden Widerstand gehen von hier zu Arduino (zu Pin Analog 0) : Beweglich
0cm 1 2 3 4 5 6 7 8 9 10
0in 1 2 3
leitende Fläche mit, sich durch Distanz vergrößerndem, Widerstand
Widerstand zwischen Sensor und Ground muss so groß sein, wie der niedrigste Widerstand des Sensors (Spannungsteiler)

LOOPS

out of the Book:
30 Arduino Projects for the Evil Genius

YOU FIND THIS EXAMPLE ONLINE:

http://arduino.cc/en/Tutorial/ForLoop

Loops allow us to repeat a group of commands a certain number of times or until some condition is met.

DESCRIPTION

The for statement is used to repeat a block of statements enclosed in curly braces.

An increment counter (ZÄHLER) is usually used to increment and terminate the loop.

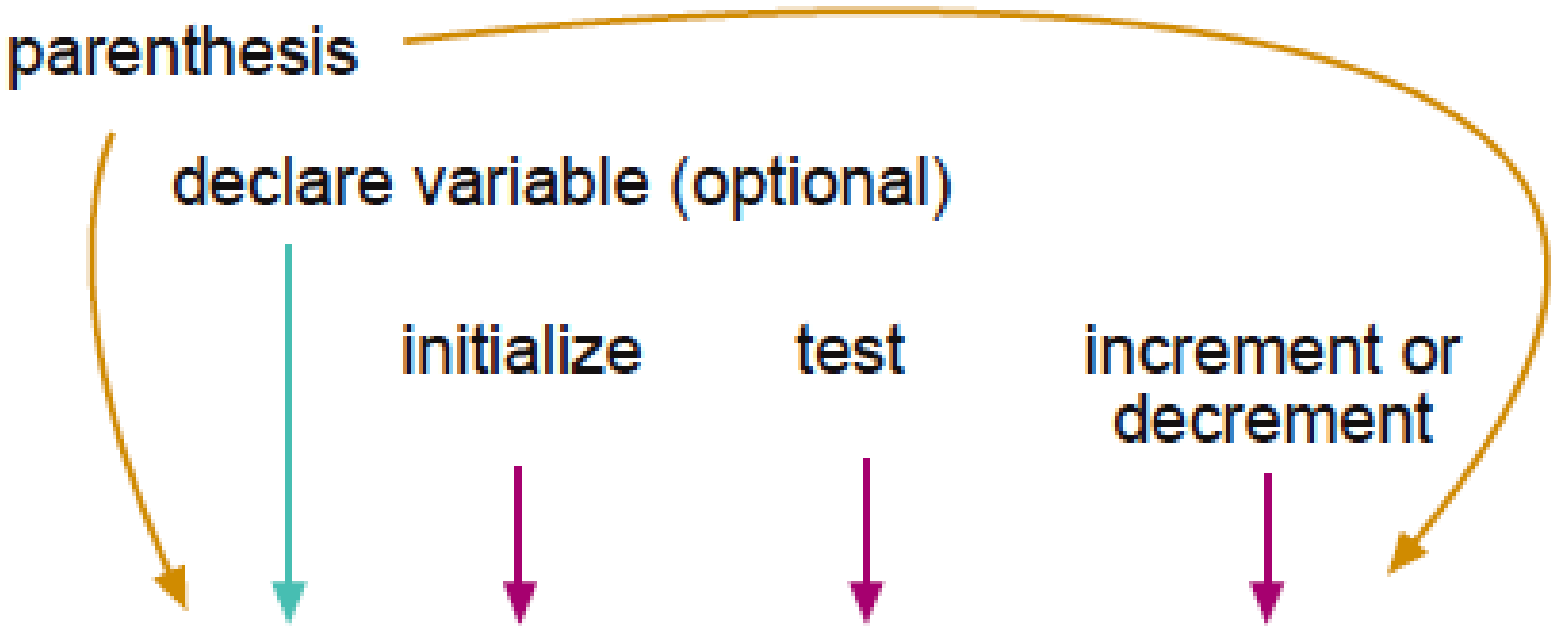
The for statement is useful for any repetitive operation,

and is often used in combination with arrays to operate on collections of data/pins.

There are three parts to the for loop header:

```
for (initialization; condition; increment) {

//statement(s);

}
```

SYNTAX OF A FOR LOOP




```
for(int x = 0; x < 100; x++){

    println(x);  // prints 0 to 99
}
```

The initialization happens first and exactly once.

Each time through the loop, the condition is tested;

if it's true, the statement block, and the increment is executed, then the condition is tested again.

When the condition becomes false, the loop ends.

Another example, fade an LED up and down with one for loop:

```
void loop()
{
   int x = 1;
   for (int i = 0; i > -1; i = i + x){
      analogWrite(PWMpin, i);
      if (i == 255) x = -1;            // switch direction at peak
      delay(10);
   }
}
```

```
// loop from the lowest pin to the highest:
for (int thisPin = 2; thisPin < 8; thisPin++) {
  // turn the pin on:
  digitalWrite(thisPin, HIGH);
  delay(timer);
  // turn the pin off:
  digitalWrite(thisPin, LOW);
}
```

Another way of looping in C is to use the while command.

```
while (digitalRead(buttonPin) == HIGH) {
  calibrate();
}
```

WHILE COMMAND

DESCRIPTION

while loops will loop continuously,

and infinitely,

until the expression inside the parenthesis, () becomes false.

Something must change the tested variable, or the while loop will never exit.

This could be in your code, such as an incremented variable, or an external condition, such as testing a sensor.

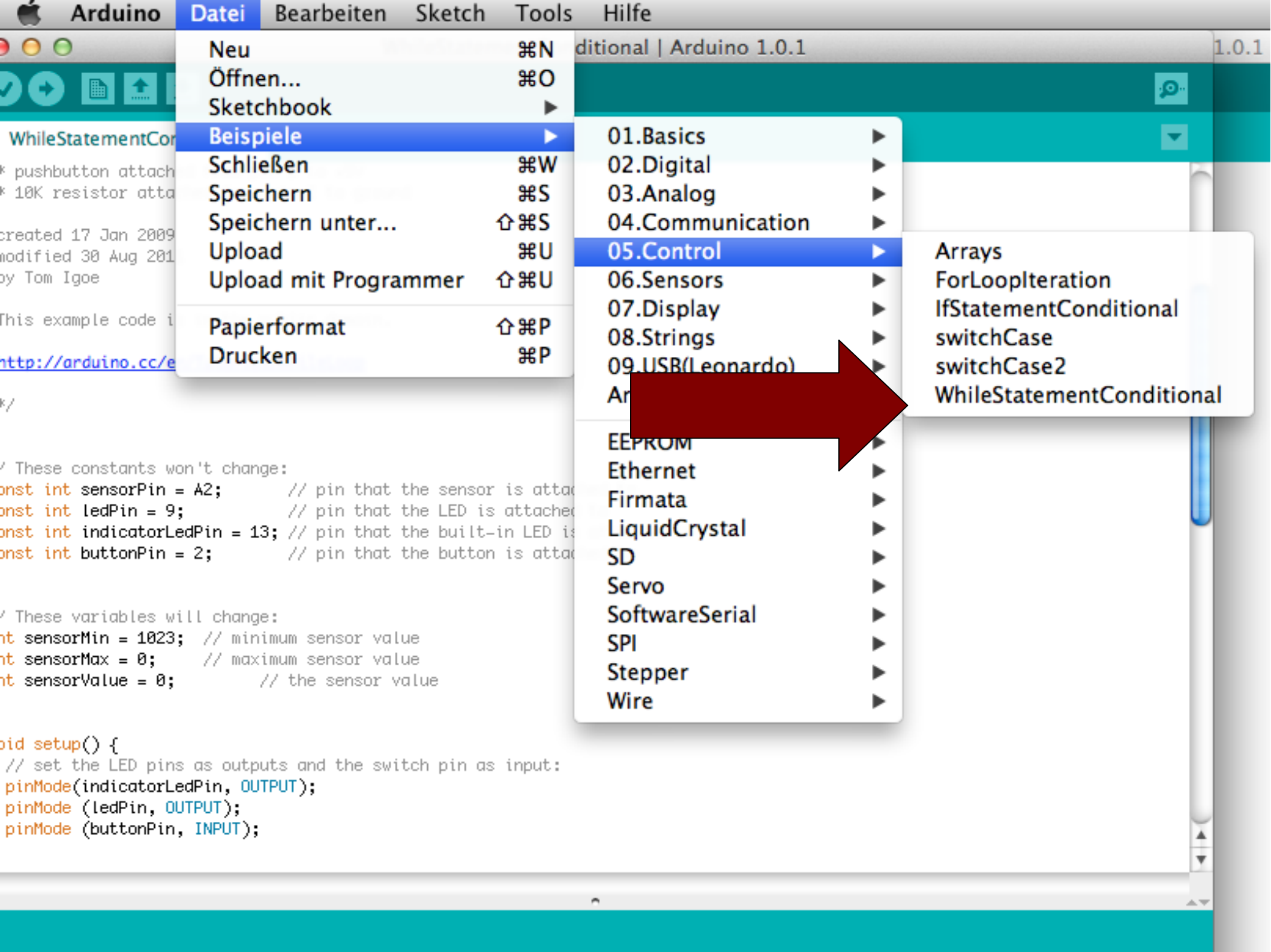
SYNTAX

```
while(expression){
  // statement(s)
}
```

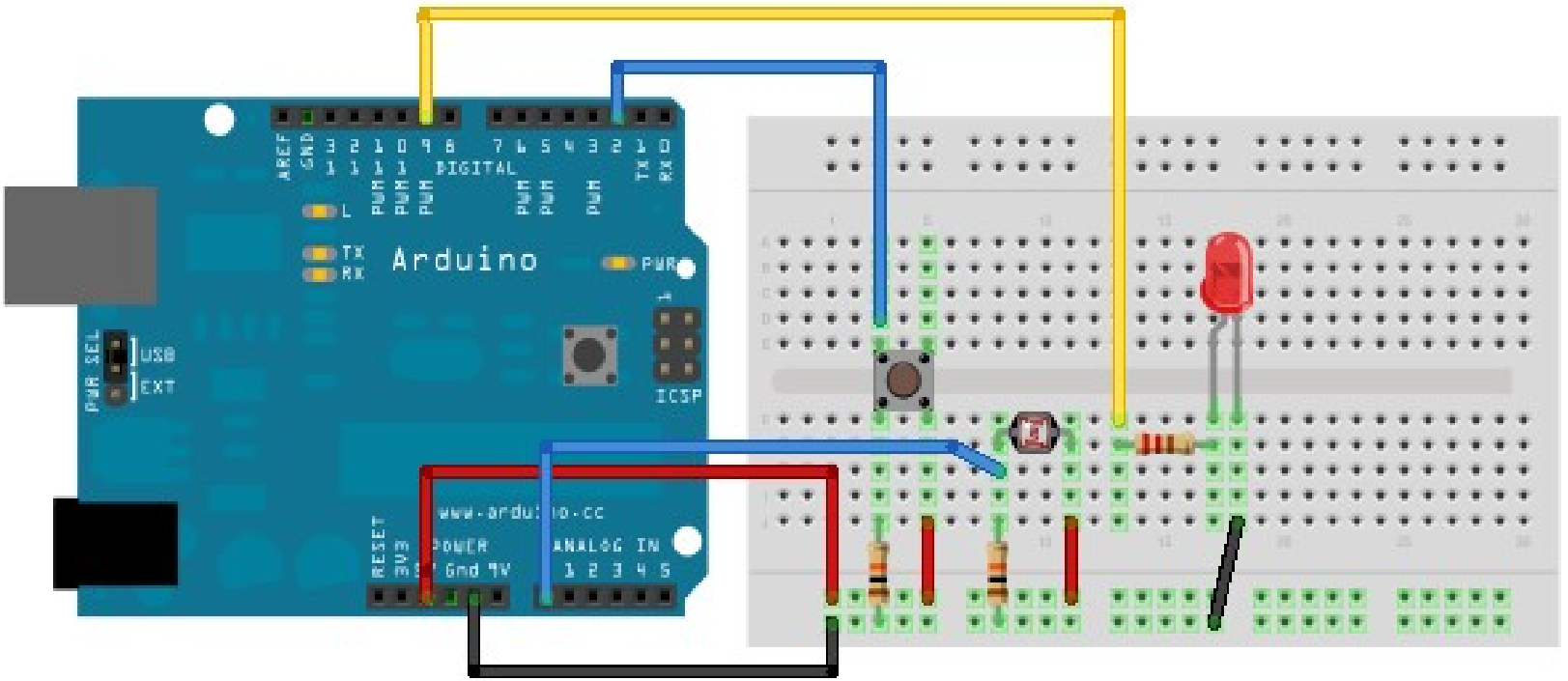
While your are in my house {

You can eat as much chocolate as you like

}


Arduino
Datei
Bearbeiten
Sketch
Tools
Hilfe
Neu ⌘N
Öffnen... ⌘O
Sketchbook
Beispiele
Schließen ⌘W
Speichern ⌘S
Speichern unter... ⇧⌘S
Upload ⌘U
Upload mit Programmer ⇧⌘U
Papierformat ⇧⌘P
Drucken ⌘P
01.Basics
02.Digital
03.Analog
04.Communication
05.Control
06.Sensors
07.Display
08.Strings
09.USB(Leonardo)
EEPROM
Ethernet
Firmata
LiquidCrystal
SD
Servo
SoftwareSerial
SPI
Stepper
Wire
Arrays
ForLoopIteration
IfStatementConditional
switchCase
switchCase2
WhileStatementConditional
ditional | Arduino 1.0.1
1.0.1
WhileStatementCor
* pushbutton attach
* 10K resistor atta
created 17 Jan 2009
modified 30 Aug 201
by Tom Igoe
This example code i
http://arduino.cc/e
*/
/ These constants won't change:
onst int sensorPin = A2; // pin that the sensor is atta
onst int ledPin = 9; // pin that the LED is attache
onst int indicatorLedPin = 13; // pin that the built-in LED i
onst int buttonPin = 2; // pin that the button is atta
/ These variables will change:
nt sensorMin = 1023; // minimum sensor value
nt sensorMax = 0; // maximum sensor value
nt sensorValue = 0; // the sensor value
oid setup() {
// set the LED pins as outputs and the switch pin as input:
pinMode(indicatorLedPin, OUTPUT);
pinMode (ledPin, OUTPUT);
pinMode (buttonPin, INPUT);

http://arduino.cc/en/Tutorial/WhileLoop



{}

The expression in parentheses after while must be true to stay in the loop. When it is no longer true, the sketch will continue running the commands after the final curly brace.

The curly braces are used to bracket together a group of commands. In programming parlance, they are known as a block.

```
// These constants won't change:
const int sensorPin = A2;       // pin that the sensor is attached to
const int ledPin = 9;           // pin that the LED is attached to
const int indicatorLedPin = 13; // pin that the built-in LED is attached to
const int buttonPin = 2;        // pin that the button is attached to


// These variables will change:
int sensorMin = 1023;  // minimum sensor value
int sensorMax = 0;     // maximum sensor value
int sensorValue = 0;         // the sensor value


void setup() {
  // set the LED pins as outputs and the switch pin as input:
  pinMode(indicatorLedPin, OUTPUT);
  pinMode (ledPin, OUTPUT);
  pinMode (buttonPin, INPUT);
}
```

```
void loop() {
  // while the button is pressed, take calibration readings:
  while (digitalRead(buttonPin) == HIGH) {
    calibrate();
  }
  // signal the end of the calibration period
  digitalWrite(indicatorLedPin, LOW);

  // read the sensor:
  sensorValue = analogRead(sensorPin);

  // apply the calibration to the sensor reading
  sensorValue = map(sensorValue, sensorMin, sensorMax, 0, 255);

  // in case the sensor value is outside the range seen during calibration
  sensorValue = constrain(sensorValue, 0, 255);

  // fade the LED using the calibrated value:
  analogWrite(ledPin, sensorValue);
}
```

```
void calibrate() {
  // turn on the indicator LED to indicate that calibration is happening:
  digitalWrite(indicatorLedPin, HIGH);
  // read the sensor:
  sensorValue = analogRead(sensorPin);

  // record the maximum sensor value
  if (sensorValue > sensorMax) {
    sensorMax = sensorValue;
  }

  // record the minimum sensor value
  if (sensorValue < sensorMin) {
    sensorMin = sensorValue;
  }
}
```

THIS IS OUR TOOL TO MEASURE OUR SELF MADE SENSORS NOW