

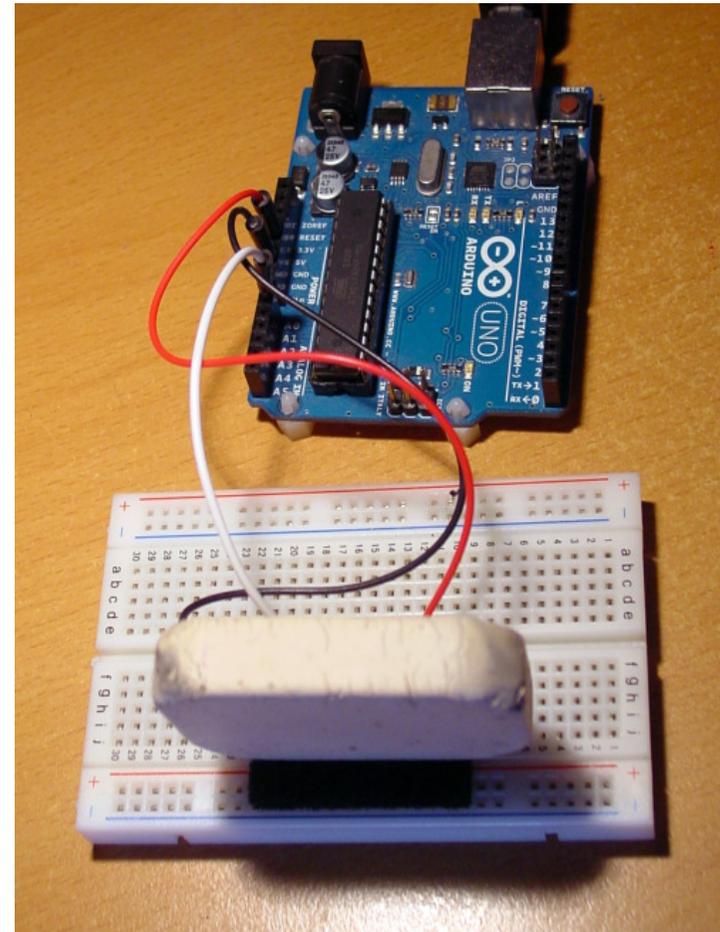
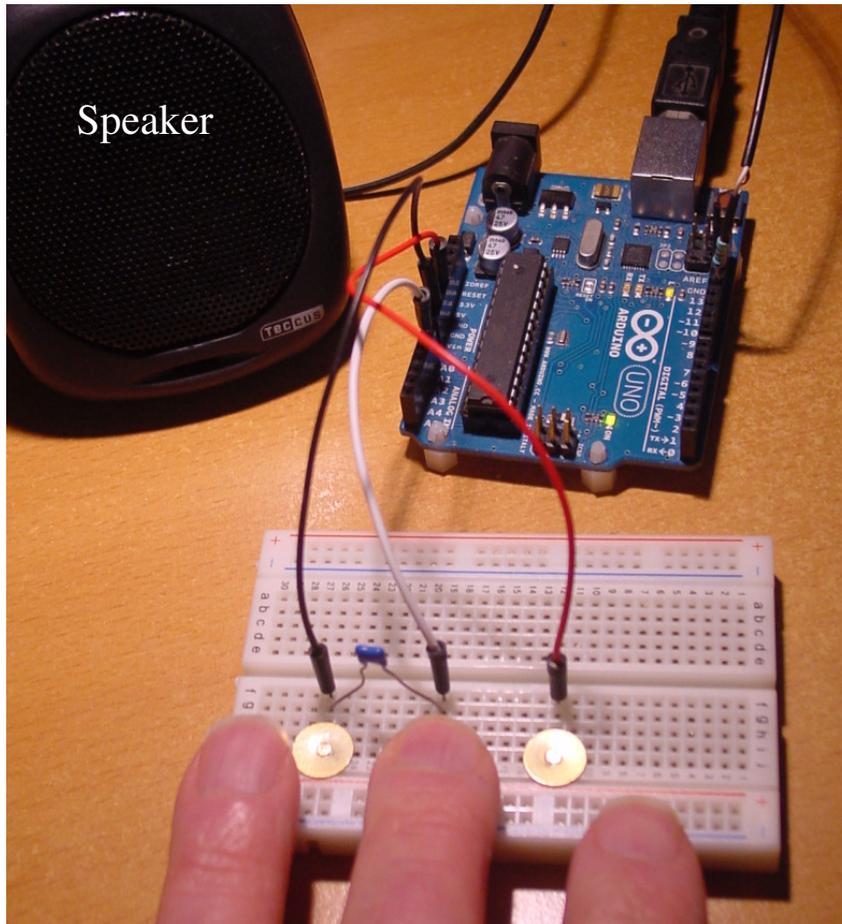
Part 0:

Workshop Arduino starter

0-Getting started (after that you can do “1” and/or “2” in any order)

1-Build a touchpad “theremin” soundbox

2-Build a pressure sensor with PC readout

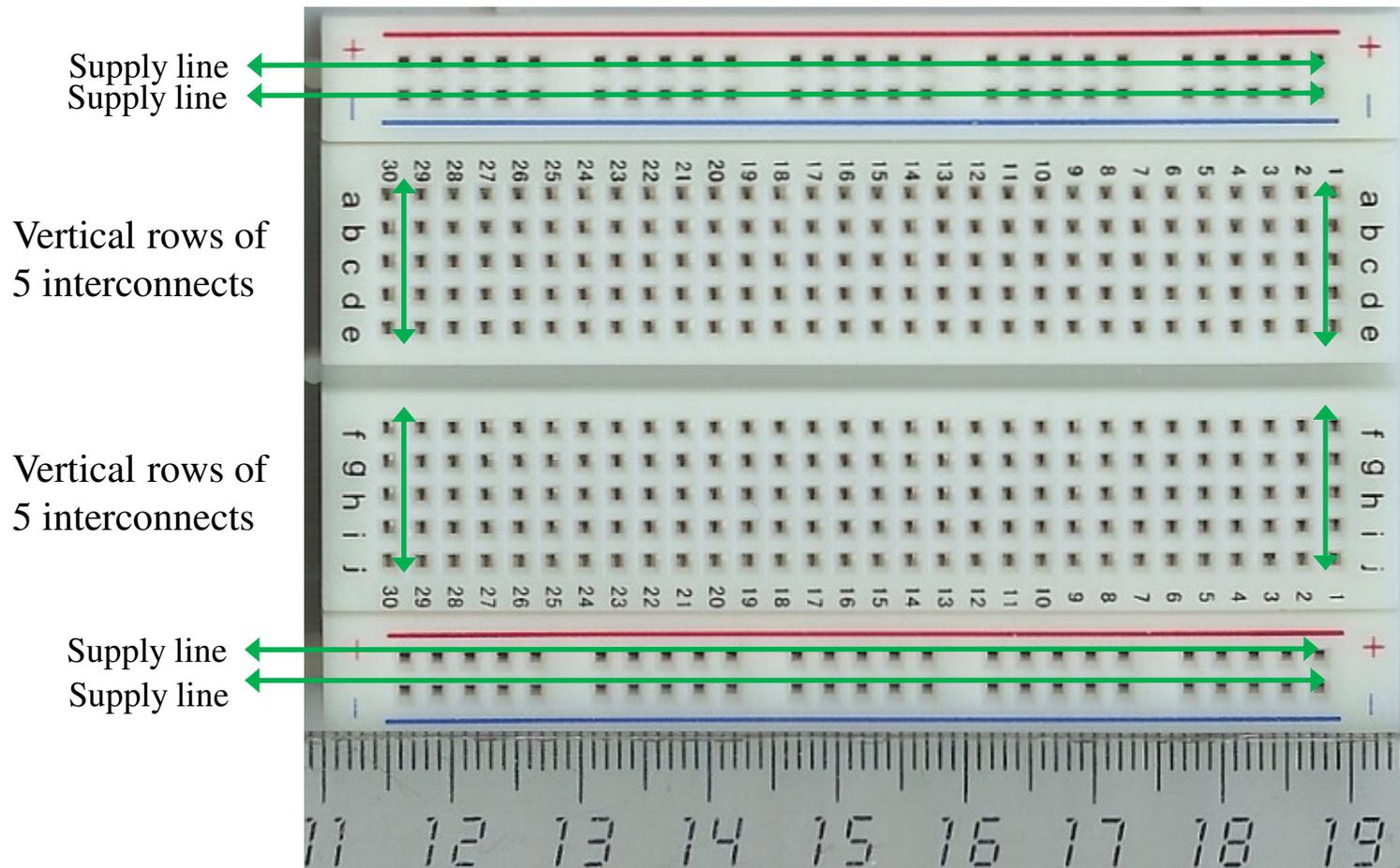


Part 0:

Getting Started

get a breadboard

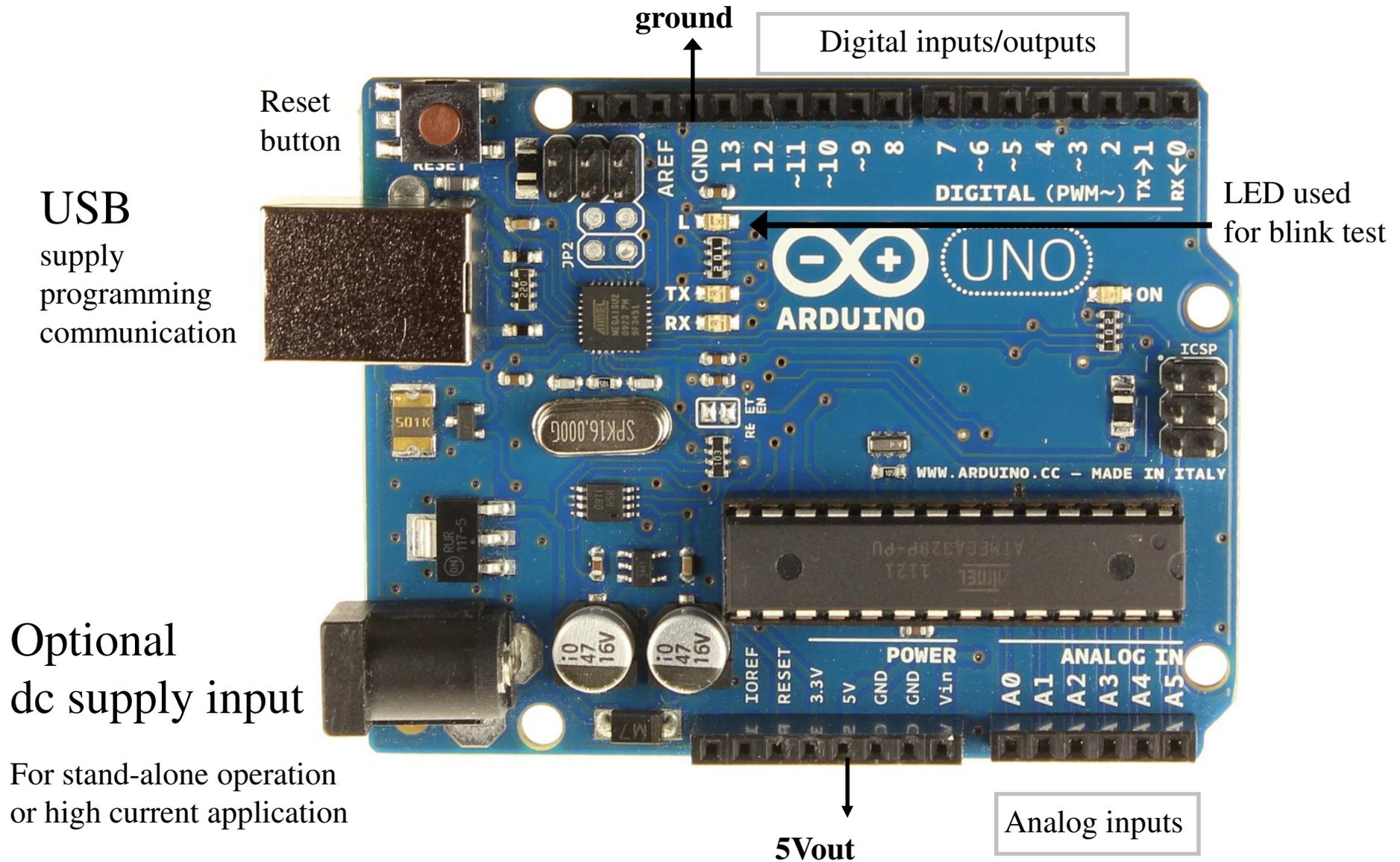
Wires and components can be plugged in and interconnected, the green arrows show the interconnection pattern of the holes



Part 0:

Getting Started

Get an Arduino board and a USB cable

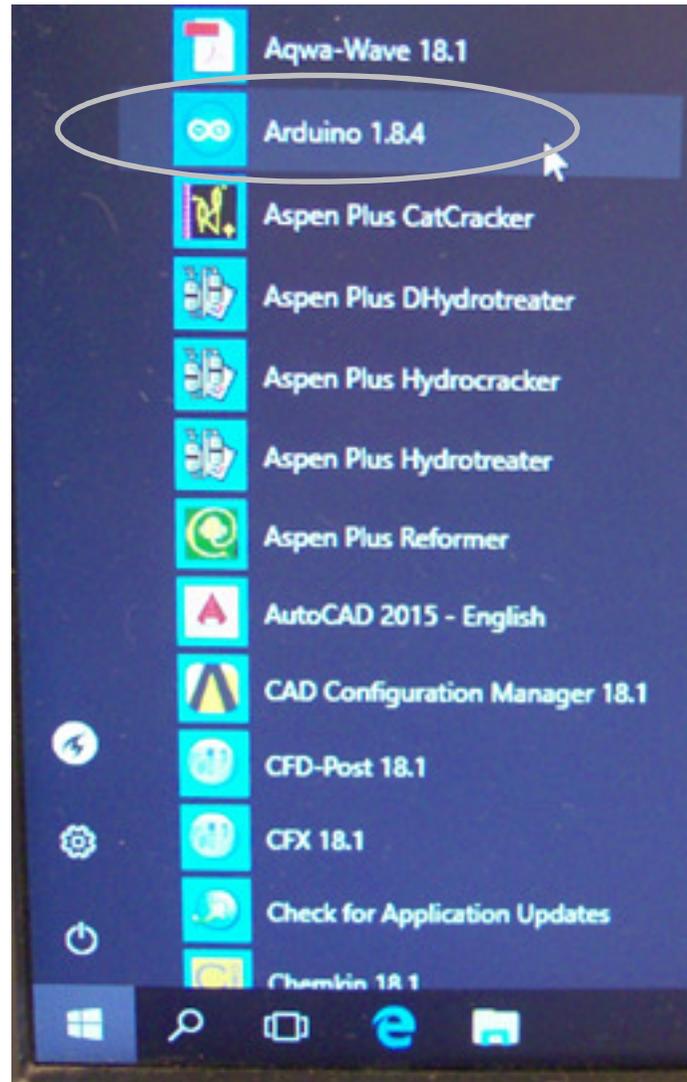


Part 0:

Getting Started

Connect the Arduino board to your PC via the USB cable

Find the software, →
start it



Concept:

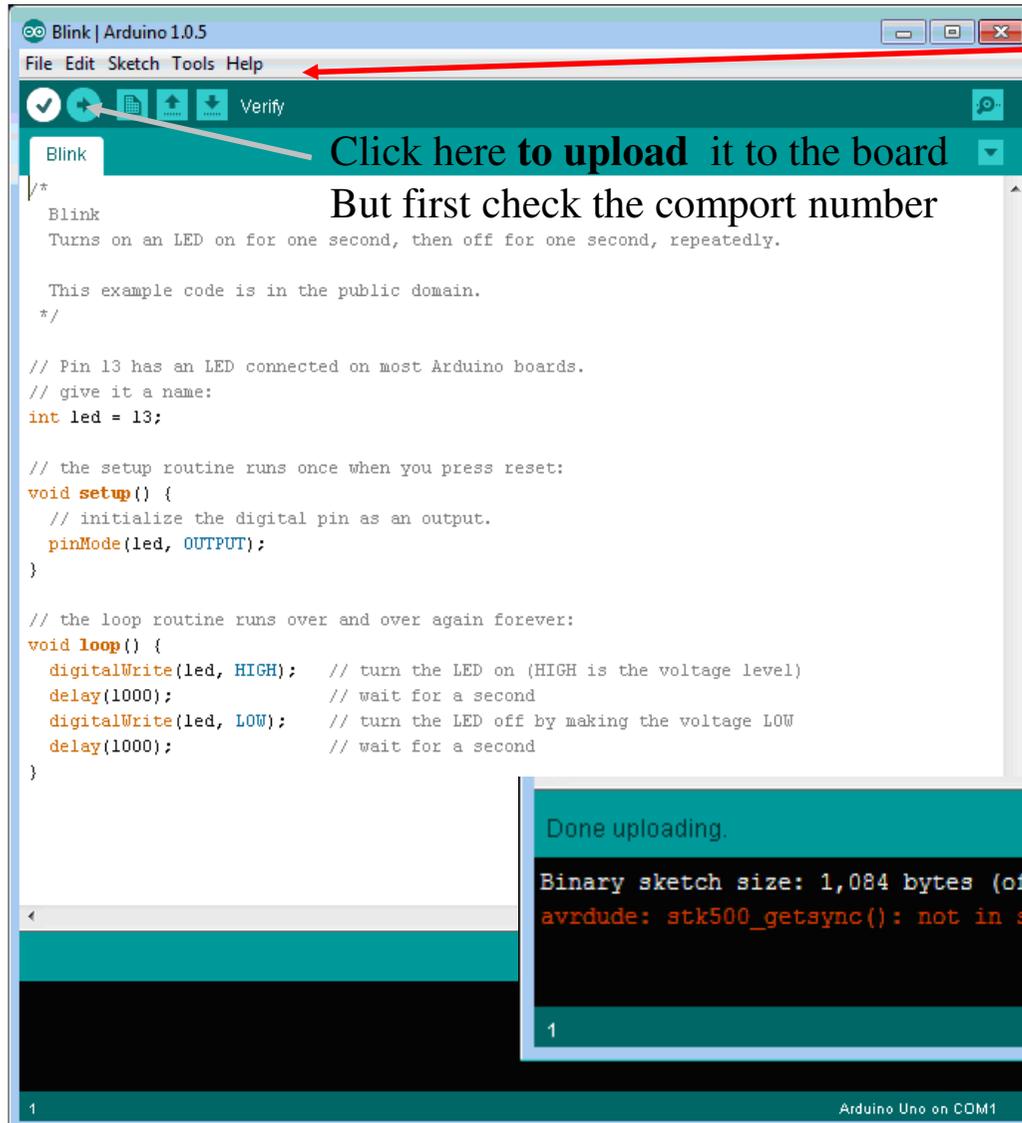
- Open the software
- Write a program (sketch)
- check & upload it to the connected Arduino board

Part 0:

Getting Started

Open a testprogram “Blink” (that should give you a blinking LED on the board)

File>Examples>Basic>Blink



The screenshot shows the Arduino IDE interface. The menu bar includes File, Edit, Sketch, Tools, and Help. The toolbar contains icons for opening files, saving, and uploading. A red arrow points to the upload button (a right-pointing arrow). Below the toolbar, a teal box contains the text: "Click here to upload it to the board" and "But first check the comport number". The main editor area displays the Blink sketch code, which includes comments and C++ code for setting up and looping the LED pin.

```
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}
```

FIRST use “Tools” to check:
The **comport** number in use.
(change it to arduino)
The **board** model
Should be “Arduino Uno”

After uploading:

IF you see this error (red message)?
Means: Wrong comport number

Done uploading.

Binary sketch size: 1,084 bytes (of a 32,256 byte maximum)
avrdude: stk500_getsync(): not in sync: resp=0x00

1

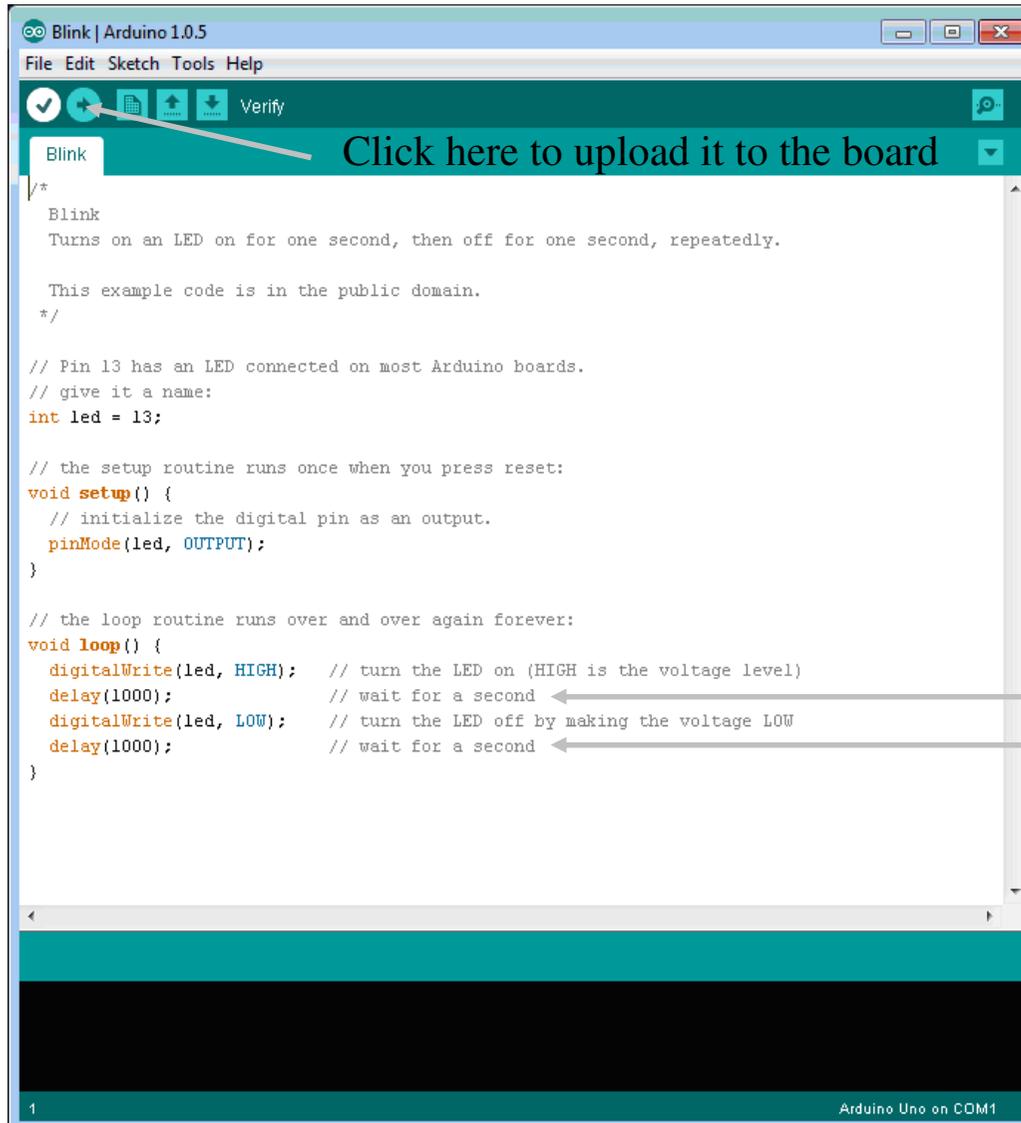
Arduino Uno on COM1

Arduino Uno on COM1

Part 0:

Getting Started

Change some parameters (and upload again) to see that everything works now



```
Blink | Arduino 1.0.5
File Edit Sketch Tools Help
[Icons] Verify
Blink Click here to upload it to the board
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}

1 Arduino Uno on COM1
```

The program has 3 parts:
1-Declare pins and variables
2-Setup conditions once
3-Main loop that “does the work”

Everything after “//” is comment text
Comment blocks between: “/*” and “*/”

Example parameter change:

Make this 50 ms and upload again

Make this 200 ms and upload

Note: the language is Caps sensitive

Part 1: Build a touchpad “theremin” soundbox

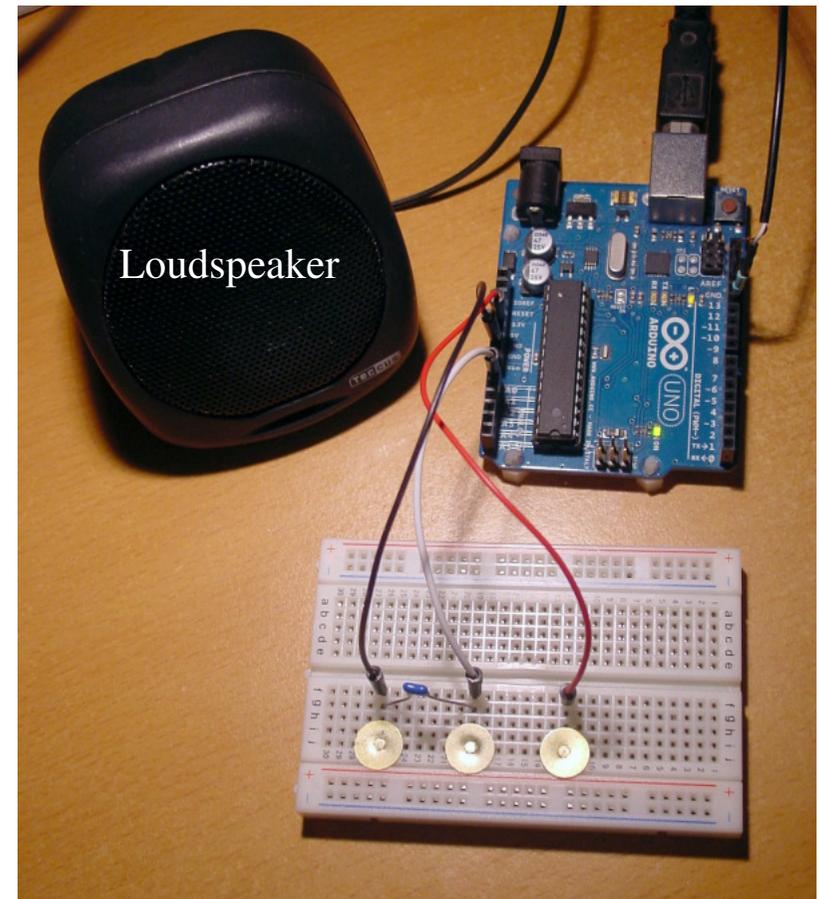
Description:

This is a basic starter project for persons that are new to electronics and/or Arduino. You can create sound with a gliding frequency that you control using the 3 touchpads (thumbtacks)

It is a very simple circuit with quite some limitations but you can build it fast and have fun using it.

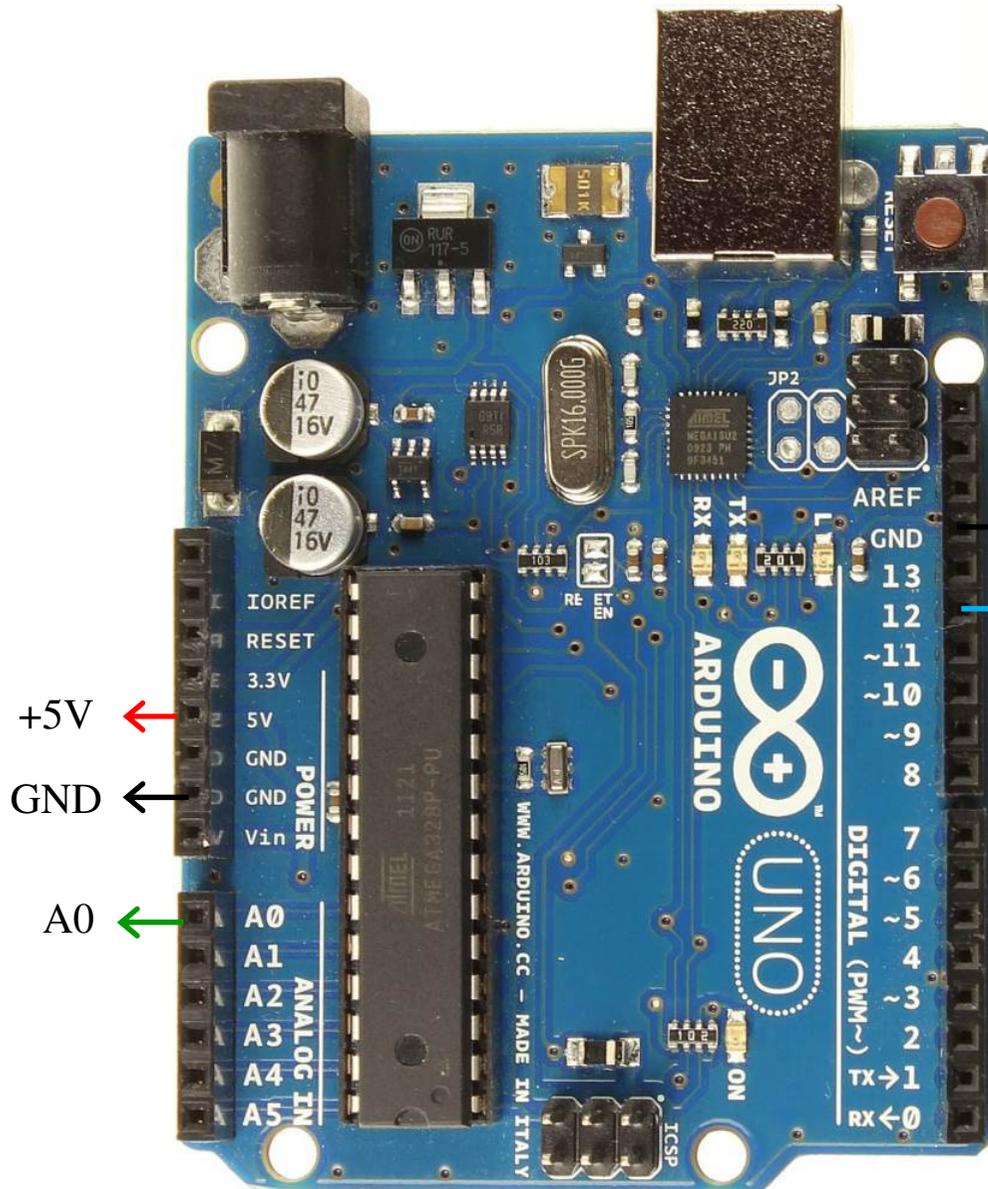
Shopping list:

- 1x Arduino Uno
- 1x breadboard
- 3x thumbtack
- 1x 10nF capacitor
- 1x loudspeaker with series resistor
- 3x patchwire (preferably red+black+other color)

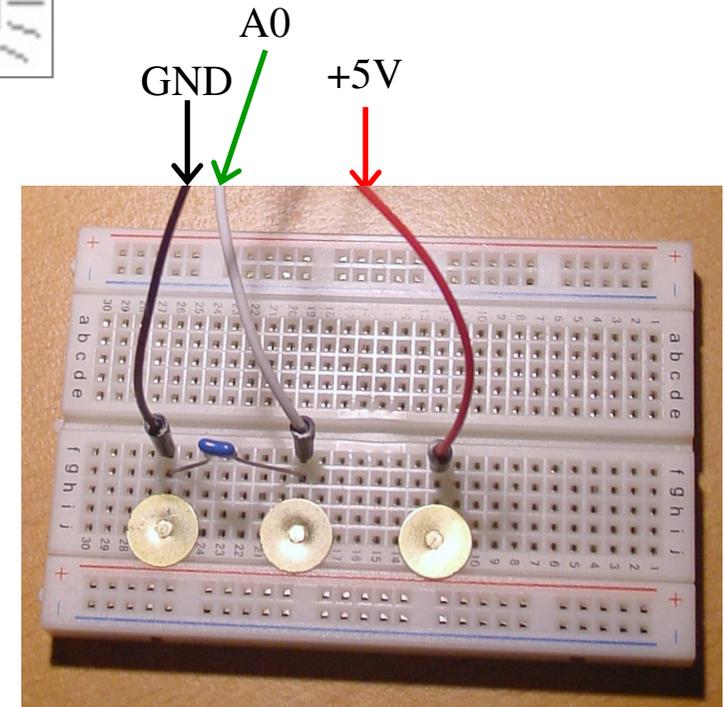
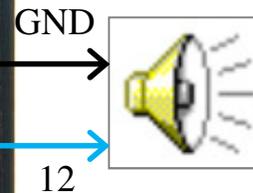


Part 1:

Build a touchpad “theremin” soundbox



- 1-Place the thumbtacks in the breadboard
- 2-Add the capacitor (A0 to GND)
- 3-Make the connections to the breadboard and to the speaker+resistor



Part 1: Build a touchpad “theremin” soundbox

How does it work: (code on next slide)

When you place one finger on the middle thumbtack (see picture) and also touch the 5V thumbtack with another finger the capacitor will be charged. The increasing voltage on the capacitor is read by the analog input pin (A0). In the Arduino code it is translated to an increasing output tone on the pin where the speaker is connected (12). If you instead touch the GND thumbtack, the capacitor will be discharged and the tone goes down. At very low voltage the tone is shut off (to give your ears some rest).

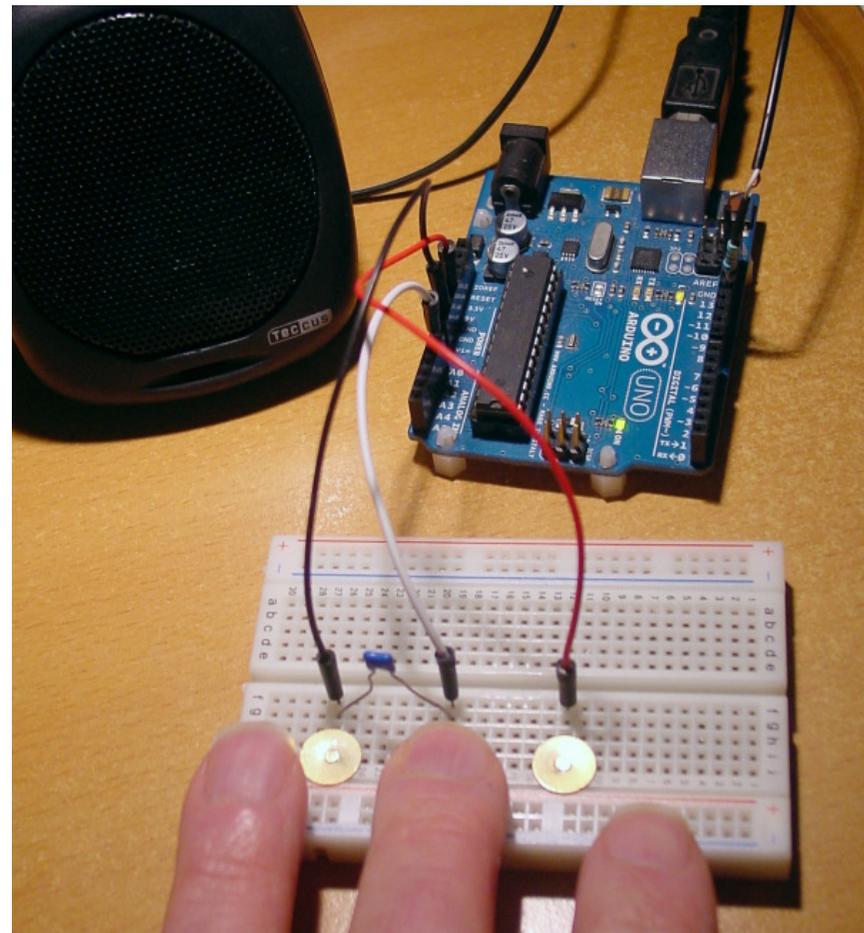
BONUS:

There is some fun physics possible here:

As the tone is set by the charge of the capacitor, you can charge/discharge also by only touching the middle thumbtack (as in the picture) and for example moving your feet over the floor, the friction creates charge and modulates the tone.

BONUS:

There is also another trick possible here: Touching only the middle thumbtack and holding your other hand close to a powerchord creates a nice vibrato by beating with 50Hz



Part 1: Build a touchpad “theremin” sandbox

The program has 3 parts:
1-Declare pins and variables
2-Setup conditions once
3-Main loop that “does the work”

This is what the program looks like in the Arduino IDE: (download at www.rs-elc.nl)
(Everything after “//” is not part of the program but are text comments)

```
// declare pins and variables first
int PadPin = A0;    // define the input pin for reading the touchpad
int SpeakerPin=12; // define the output pin for the speaker (with resistor 470 ohm in series)
int PadRead;      // variable that holds the touchpad reading
int Note;// varable that holds the frequency to be played

void setup() {}// nothing to setup here but you need this setup statement

void loop() { //start of main loop
  PadRead = analogRead(PadPin); // read voltage from Pad (0 to 1023 means 0 to 5V)
  if (PadRead>20){ // only produce sound if voltage is >0.1V
    Note=5*PadRead; // scale frequency from 100Hz to 5kHz
    tone(SpeakerPin, Note); // try also: tone(SpeakerPin, Note,5 ) for resonance sweep sound type
  }
  else{noTone (SpeakerPin);} // shut off sound if voltage too low
  delay(17); // wait 17ms
  //this specific wait time creates vibrato when touching the Pad, caused by beating with 50Hz from the environment
  // if you are in a country with 60Hz mains use delay (14)
} //end of main loop
```

Note: the language is Caps sensitive

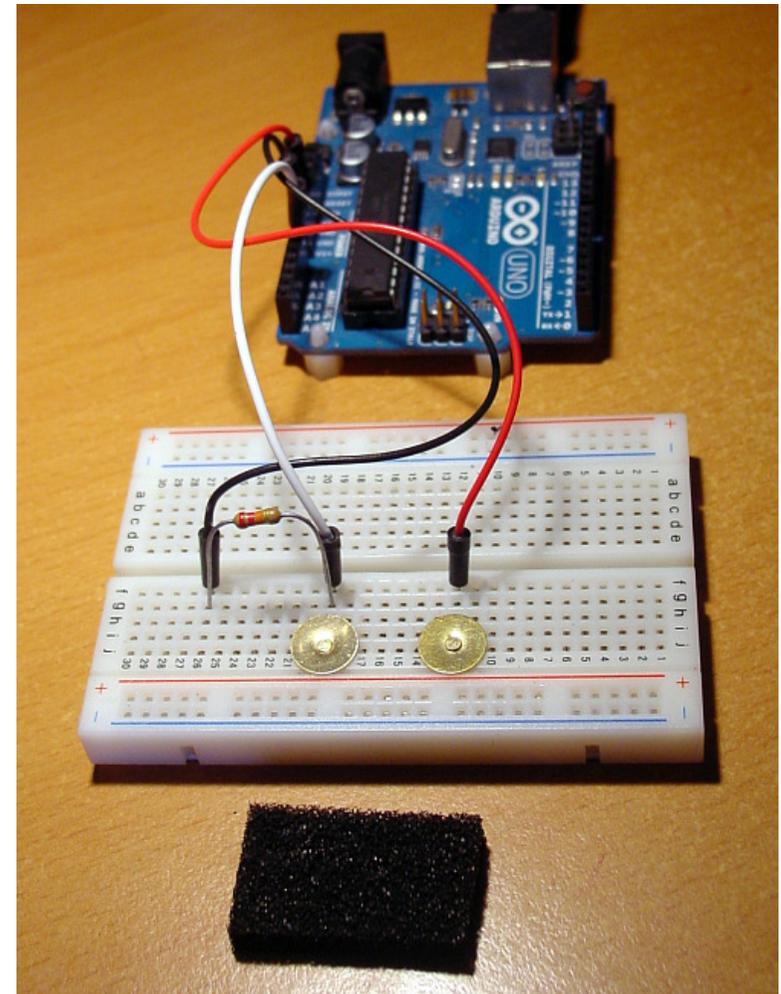
Part 2: Build a pressure sensor with PC readout

Description:

This is a basic starter project for persons that are new to electronics and/or Arduino. You will create a pressure sensor and read it's data with the PC. It is a very simple circuit with quite some limitations but you can build it fast and have fun using it.

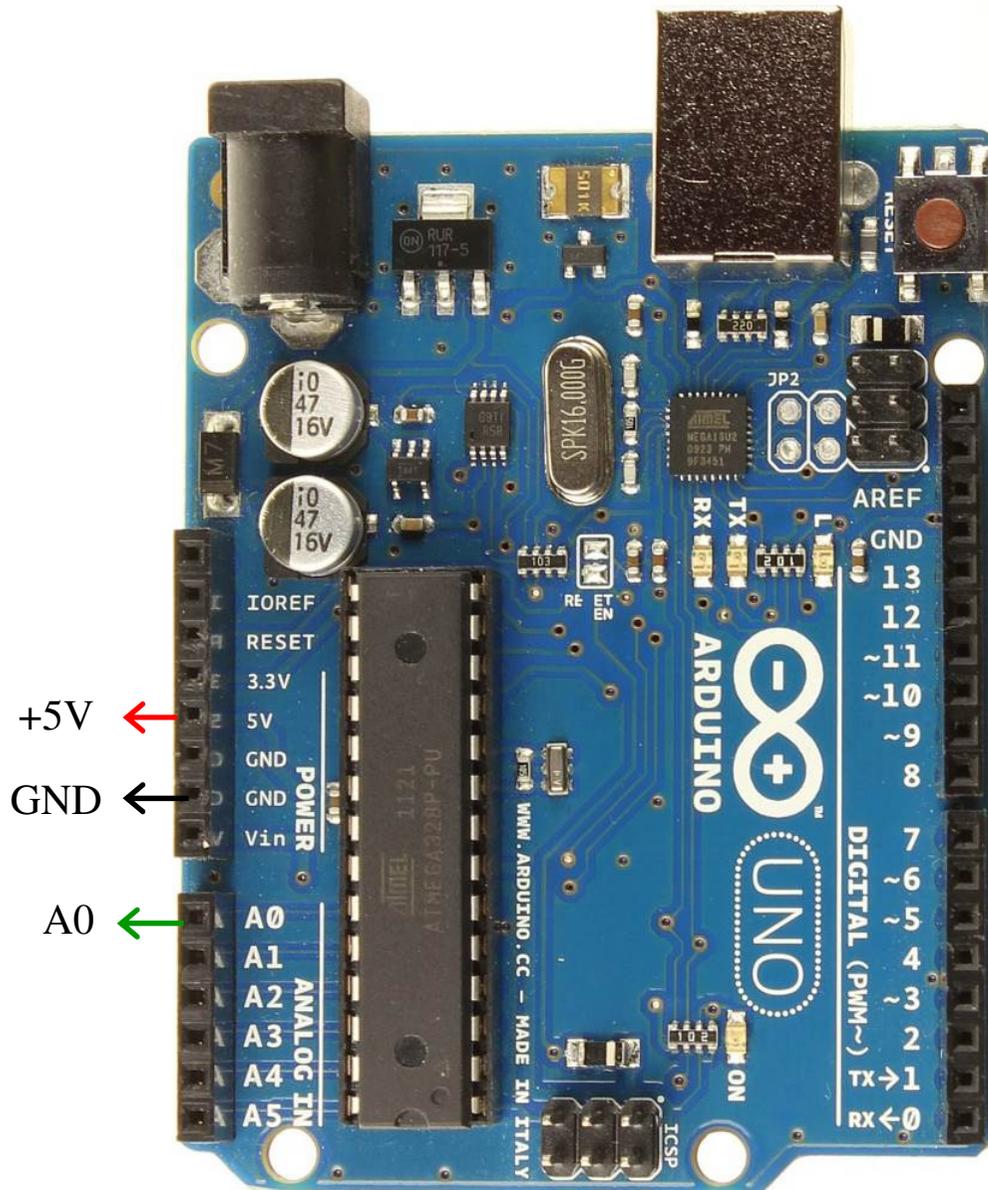
Shopping list:

- 1x Arduino Uno
- 1x breadboard
- 2x thumbtack
- 1x 22k resistor
- 1x piece of conductive foam (black)
- 3x patchwire (preferably red+black+other color)

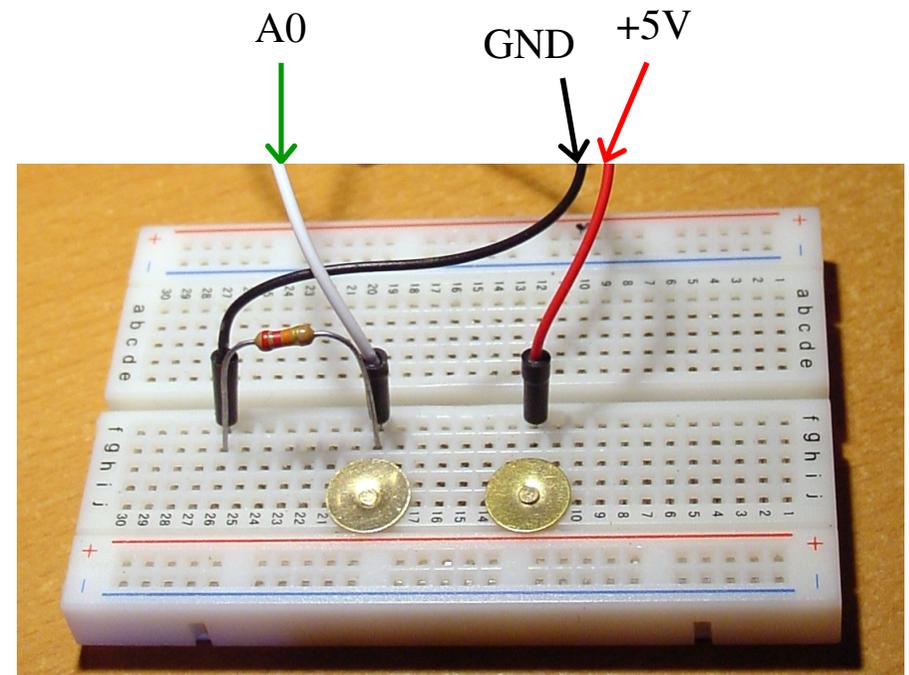


Part 2:

Build a pressure sensor with PC readout



- 1-Place the thumbtacks in the breadbord
- 2-Add the 22k resistor (A0 to GND)
- 3-Make connections to the breadboard



Part 2: Build a pressure sensor with PC readout

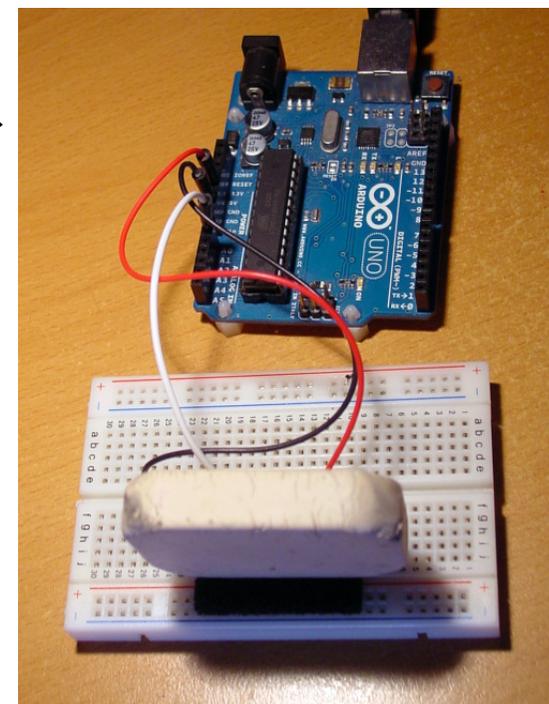
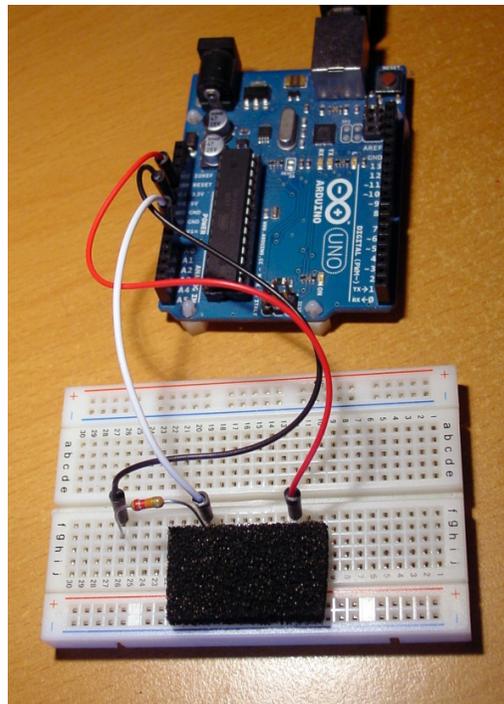
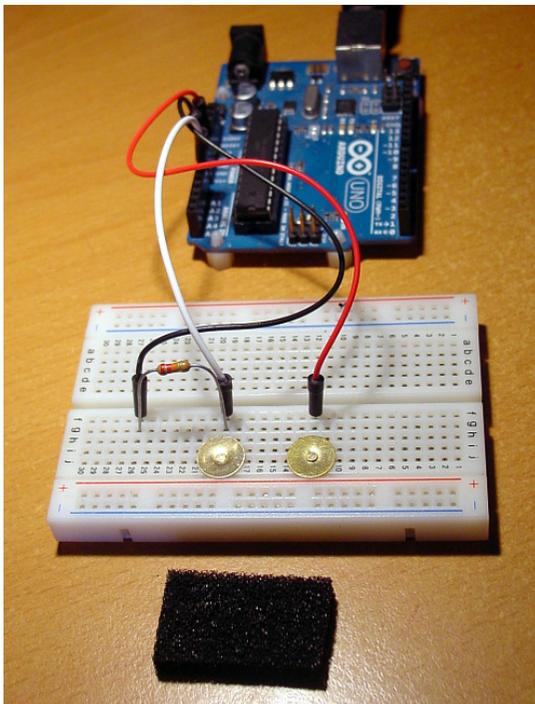
How does it work: (code on next slide)

The conductive foam (packaging material for sensitive electronics) is loaded with carbon.

When you compress it, conductance increases. In this setup it results in more current through the 22k resistor and thereby an increasing voltage read by analog input (A0) and send to the PC

Place the foam on the thumbtacks

Push or add an object



Part 2: Build a pressure sensor with PC readout

The program has 3 parts:

- 1-Declare pins and variables
- 2-Setup conditions once
- 3-Main loop that “does the work”

This is what the program looks like in the Arduino IDE: (download at www.rs-elc.nl)
(Everything after “//” is not part of the program but are text comments)

```
// declare pins and variables first
int PadPin = A0;    // define the input pin for reading the analog voltage from the touchpad
int PadRead;      // variable that holds the touchpad reading

void setup() { // setup runs once
  Serial.begin(57600); // open USB virtual comport for data with baudrate setting 57600
}

void loop() { // start of main loop that is always running
  PadRead = analogRead(PadPin); // read voltage from Pad (0 to 1023 means 0 to 5V)
  Serial.println(PadRead); // send the value to the USB virtual comport
  delay(100); // wait 100ms
} // end of main loop
```

Note: the language is Caps sensitive

Part 2:

Build a pressure sensor with PC readout

Using the pressure sensor as a “weight scale”
Reading the serial plotter while changing objects
(see: Tools/serial plotter)

