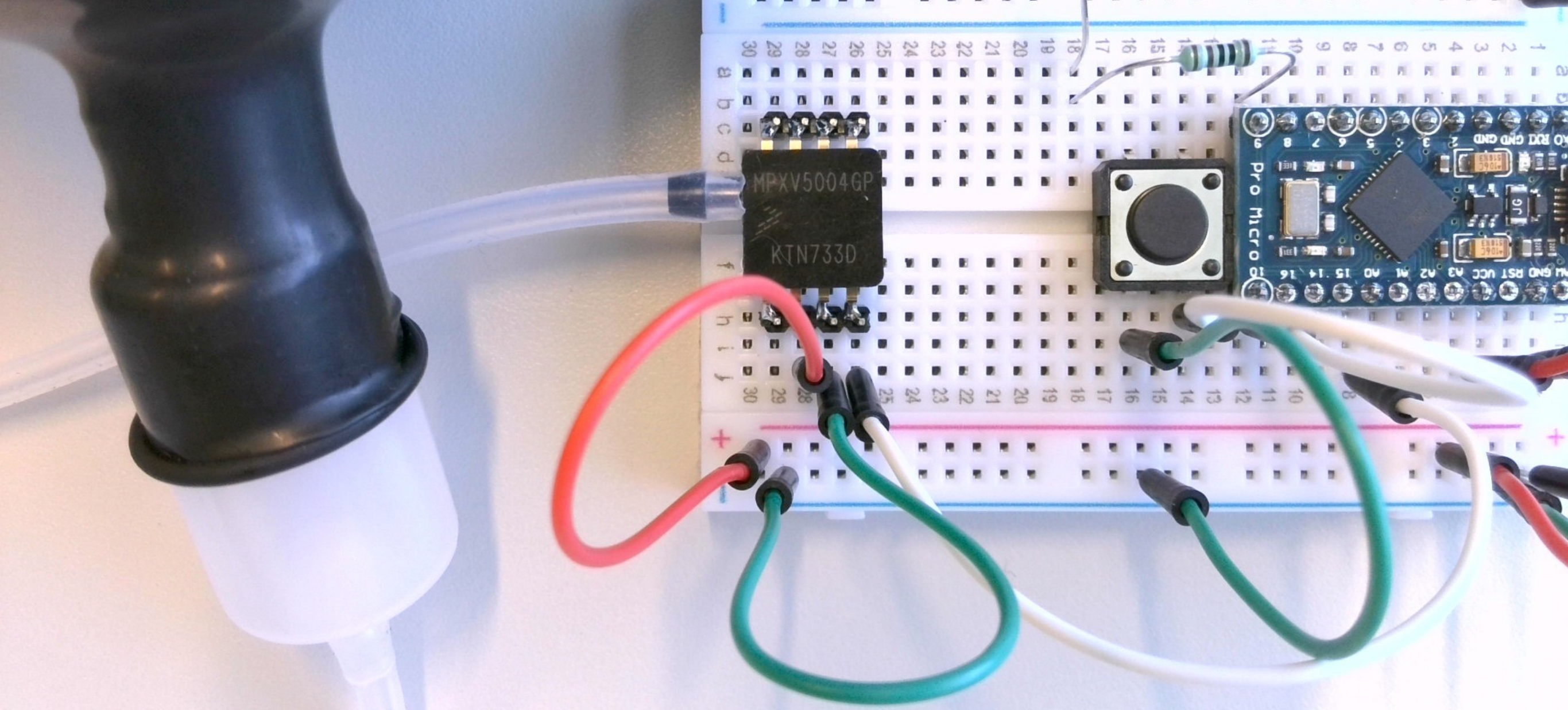


DEFORMATION

UWTMC

Matt Borland - 2019



WE'RE MAKING MUSICAL BALLOONS THAT USE DEFORMATION TO CREATE MUSIC

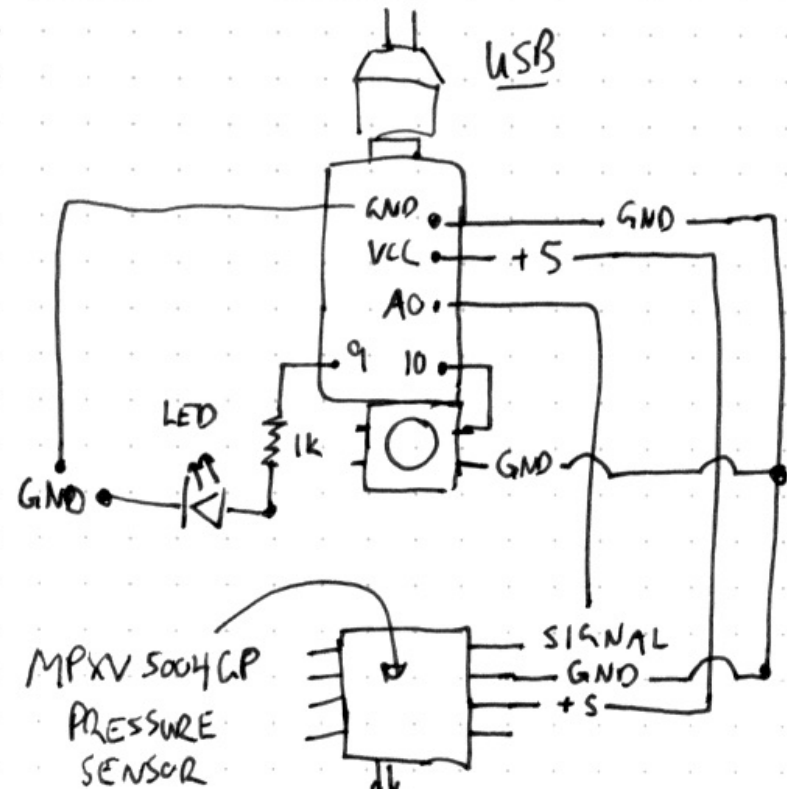
.....

We'll use sensors to measure RATE of deformation and AMOUNT of deformation, then convert those to MIDI messages which your computer will then turn into musical sounds!

Check out a fancier version of this here: <https://vimeo.com/326637479>

UWTMC DEFORMATION

①

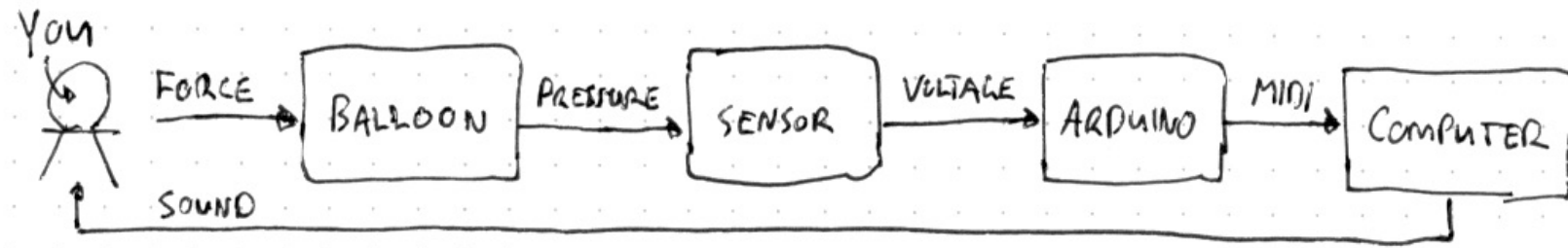


BE CAREFUL!
HOOKING THIS
UP WRONG WILL
RUIN THE SENSOR!

MEASURES
"GAUGE PRESSURE"

$$P_{\text{Gauge}} = P_{\text{BALLOON}} - P_{\text{ATMOSPHERIC}}$$

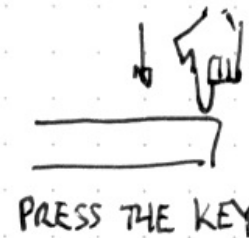
CONVERTED TO AN ANALOG
VOLTAGE (0-5V) READ BY
THE ARDUINO AND TURNED
INTO VALUES (0-1023) WHICH
ARE USED TO DECIDE
WHICH MIDI MESSAGES TO SEND



MIDI: MUSICAL INSTRUMENT DIGITAL INTERFACE

WE SEND TWO TYPES OF MESSAGES TO OUR
COMPUTERS TO BE TURNED INTO SOUNDS.

① NOTE ON

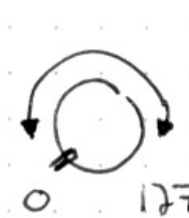


NOTE OFF



CHANNEL
1-16
VELOCITY
0-127
PITCH
0-127

② CONTROL CHANGE



POSITION OF A
KNOB OR A
SLIDER.

CHANNEL , CC NUM , VALUE
1-16 , 0-127 , 0-127

128 KNOBS!

DOWNLOADS

You need two pieces of software. Both are free and multi-platform!

Download the Arduino IDE



Arduino IDE - a software platform used to program your microcontroller.

<https://www.arduino.cc/en/Main/Software>

Helm - a software synthesizer to make musical sounds with your computer.

<https://tytel.org/helm/>

FILES

Workshops this term are at CML! All sessions run 3:30-5:30pm at Critical Media Lab, located in Communitech, 151 Charles St. W., Kitchener.

Sept. 18th: Deformation - MIDI Balloons: [Deformation Files.zip](#)

Oct. 2nd: Gravity - Shruti Box

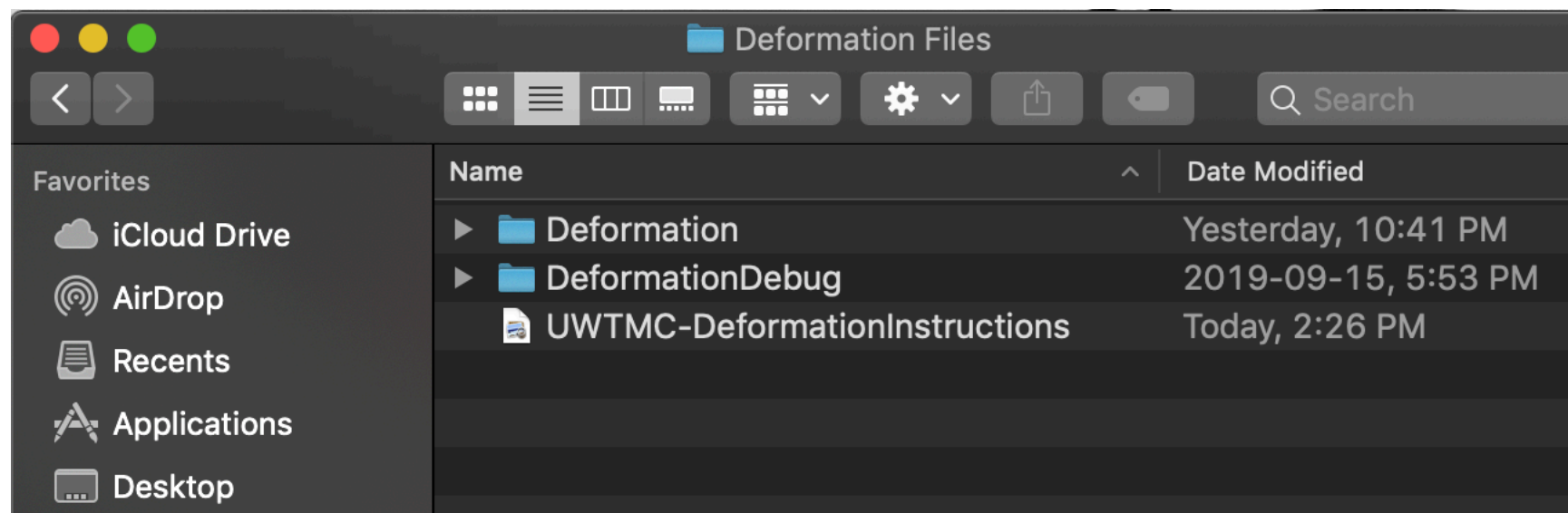
Oct. 23rd: Continuity - Seaboard and Bop Pad

Nov. 6th: Complexity - Modular Synthesis

Nov. 20th: Exploration - Co-play Patch Tables



Files are available as a ZIP at <https://uwatmc.com>

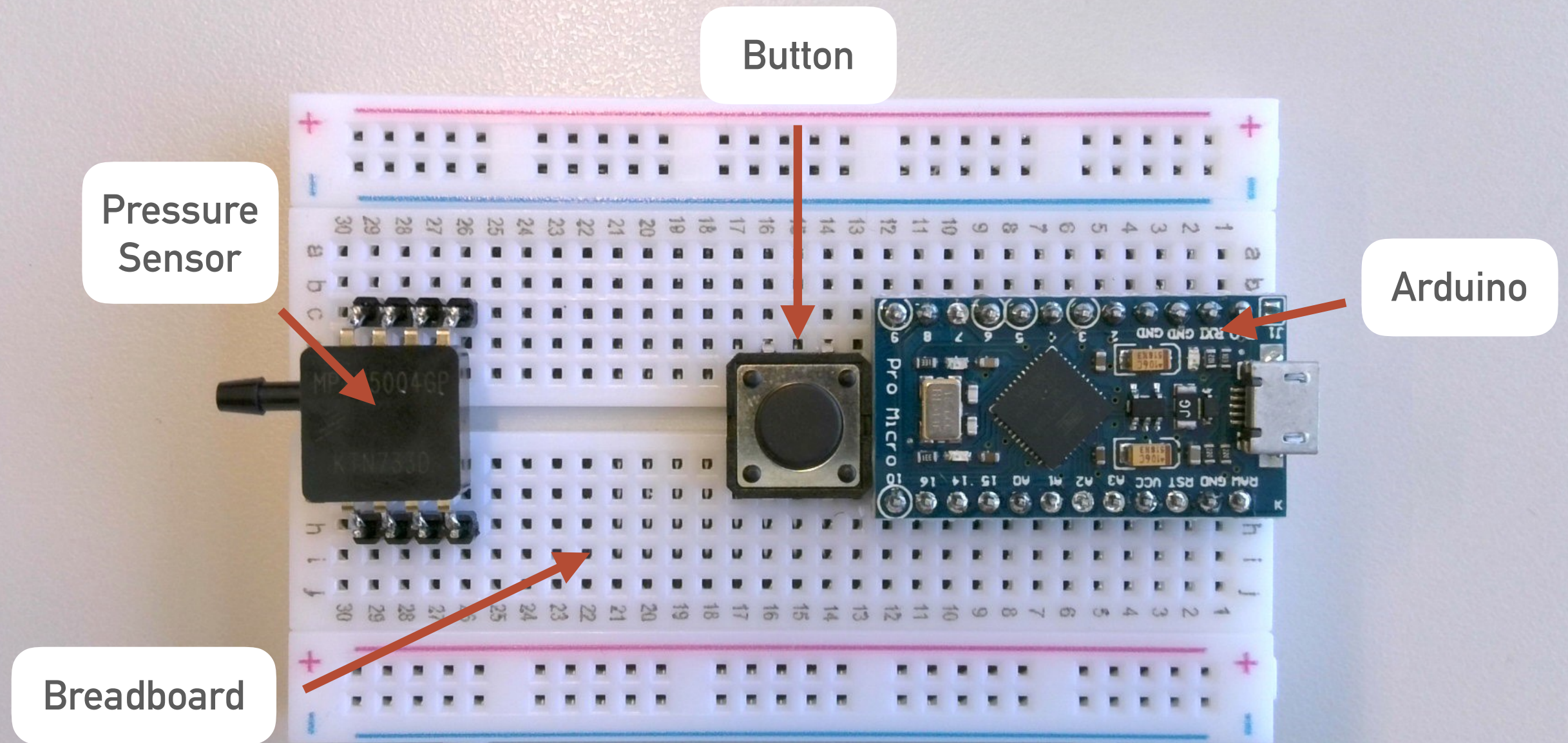


LET'S DO IT!

*Matt will explain everything,
so follow along with him on
the big screen. Ask questions if
you're unsure - other people
are probably in the same boat.*

HOOKING UP YOUR SENSORS! FOLLOW MB ON THE PROJECTOR

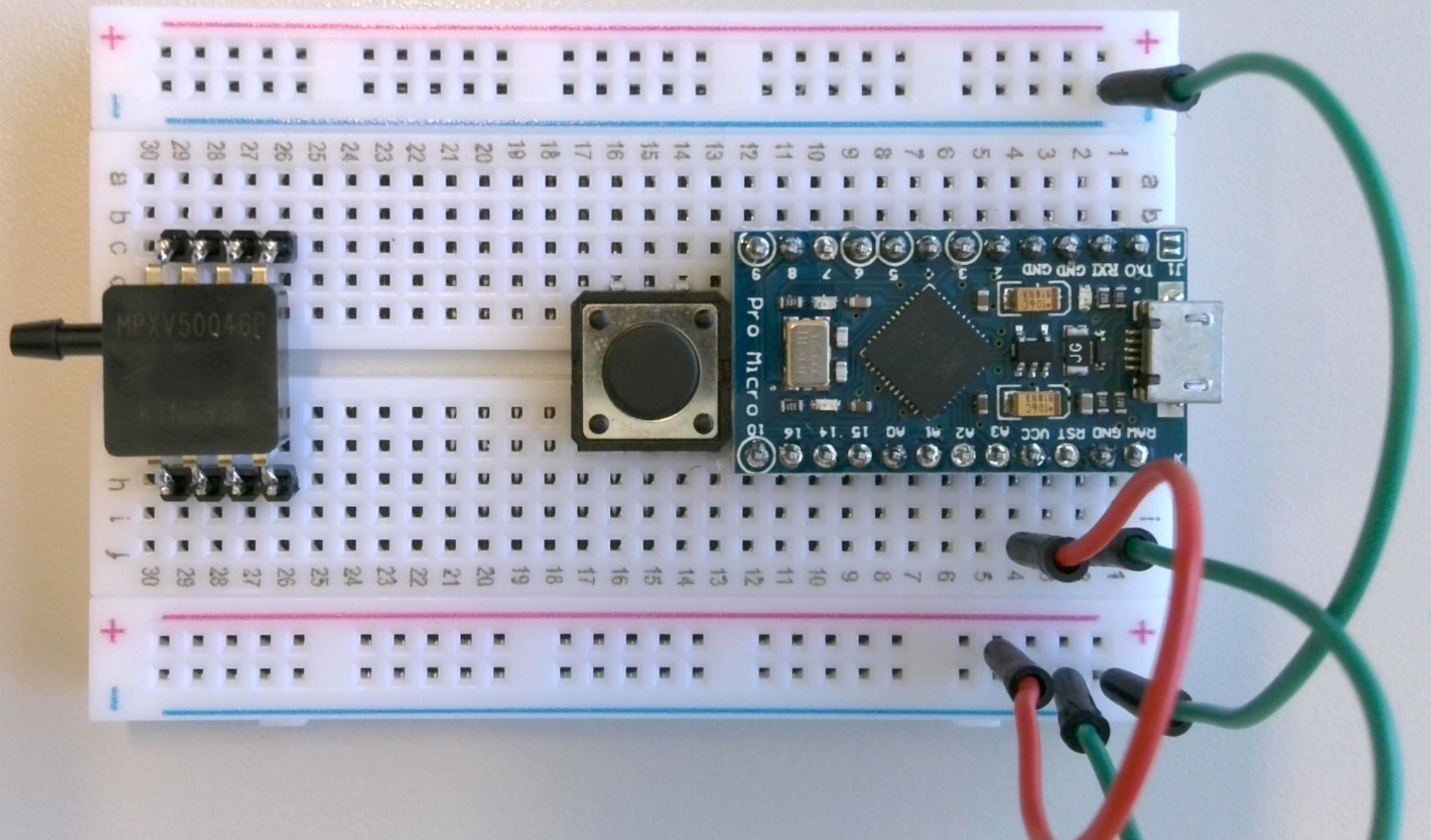
.....

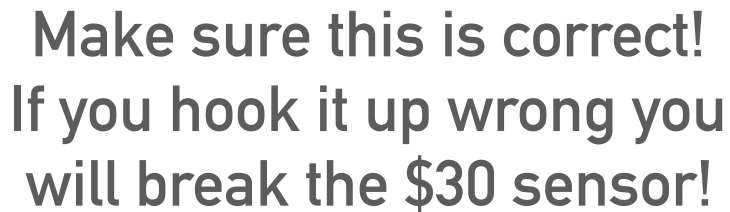


HOOKING UP YOUR SENSORS! POWER

.....

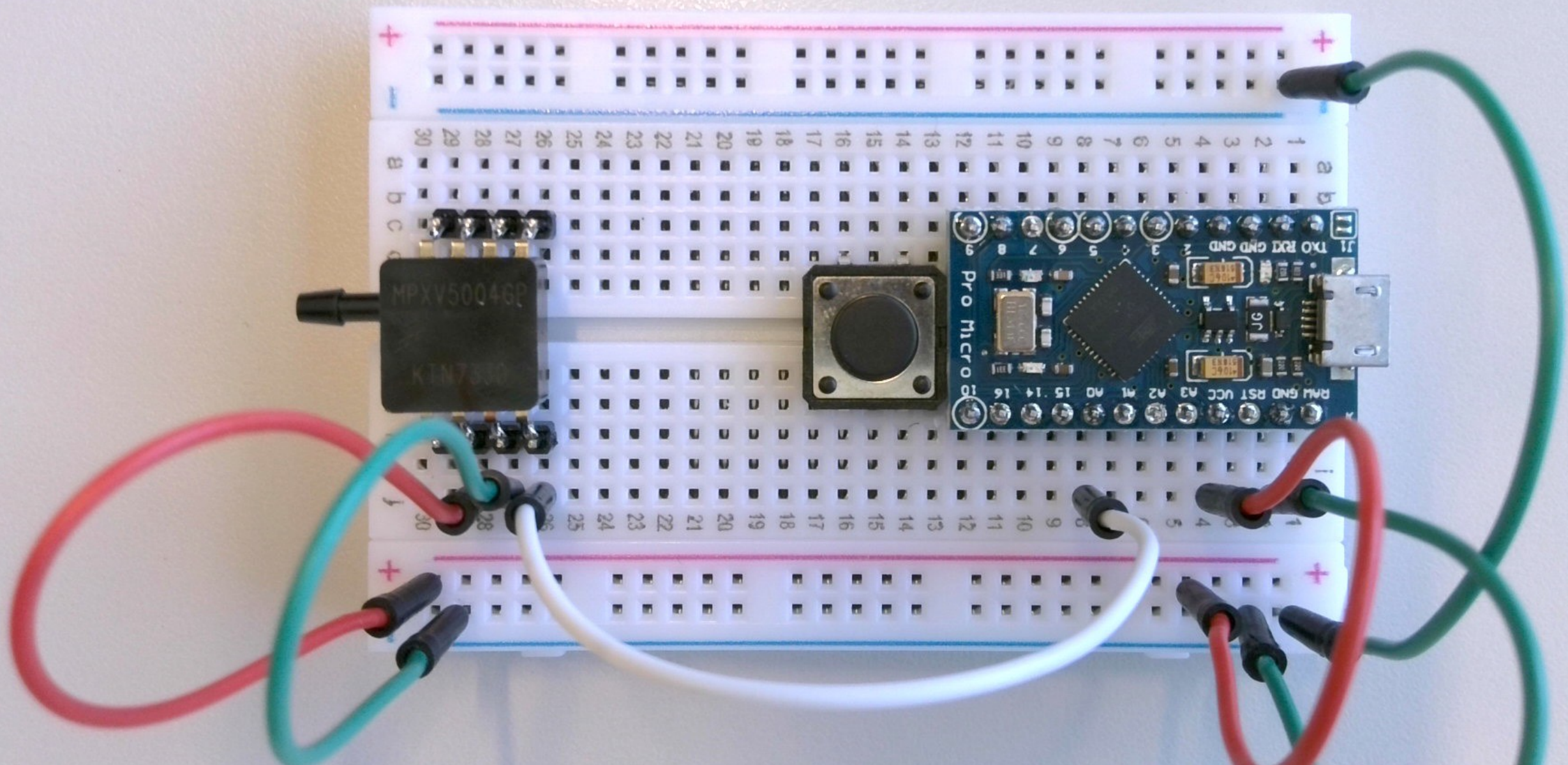
GND to the -ve Blue Row
VCC to the +ve Red Row





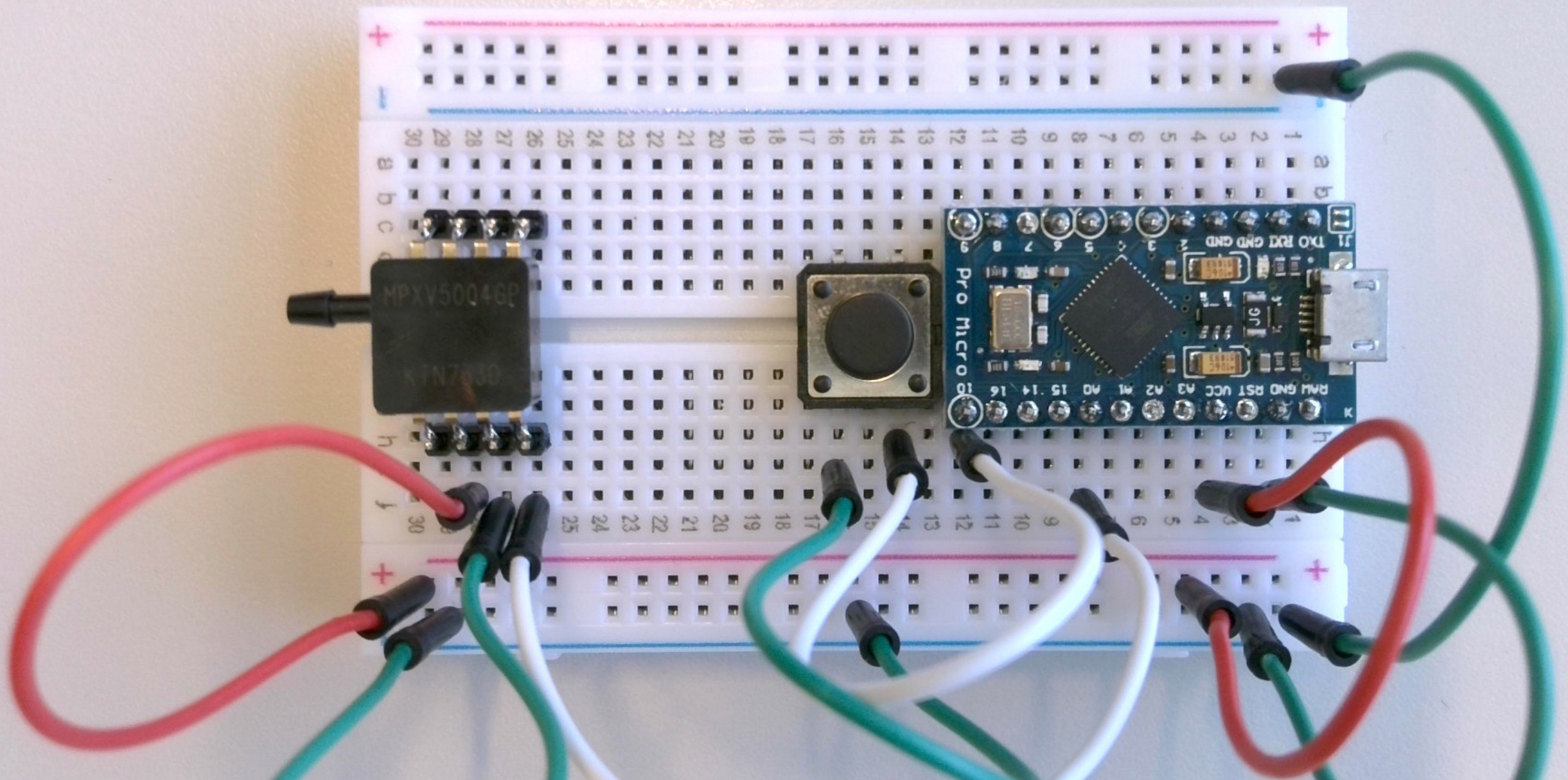
HOOKING UP YOUR SENSORS! PRESSURE SENSOR

.....



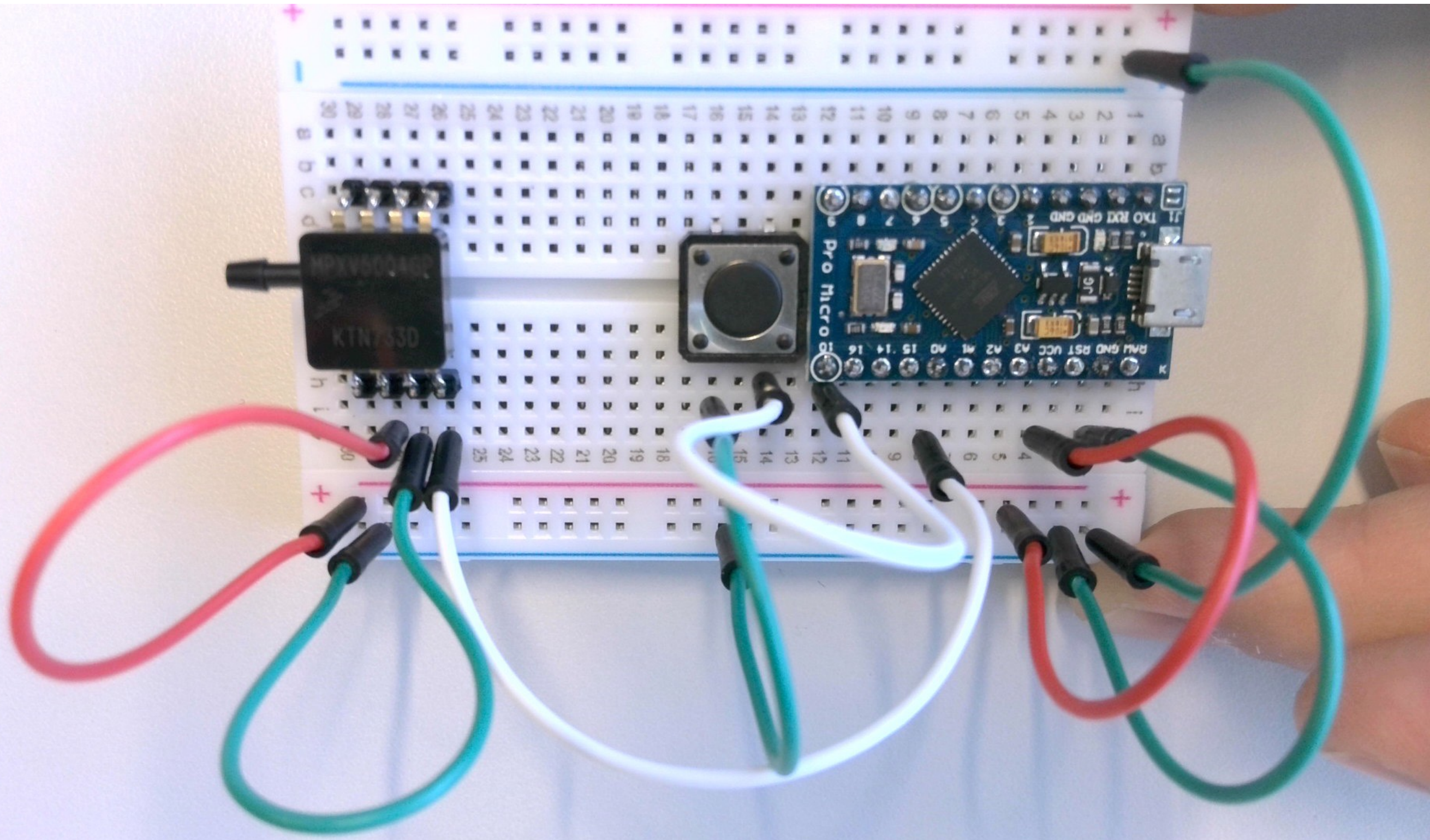
HOOKING UP YOUR SENSORS! BUTTON

.....



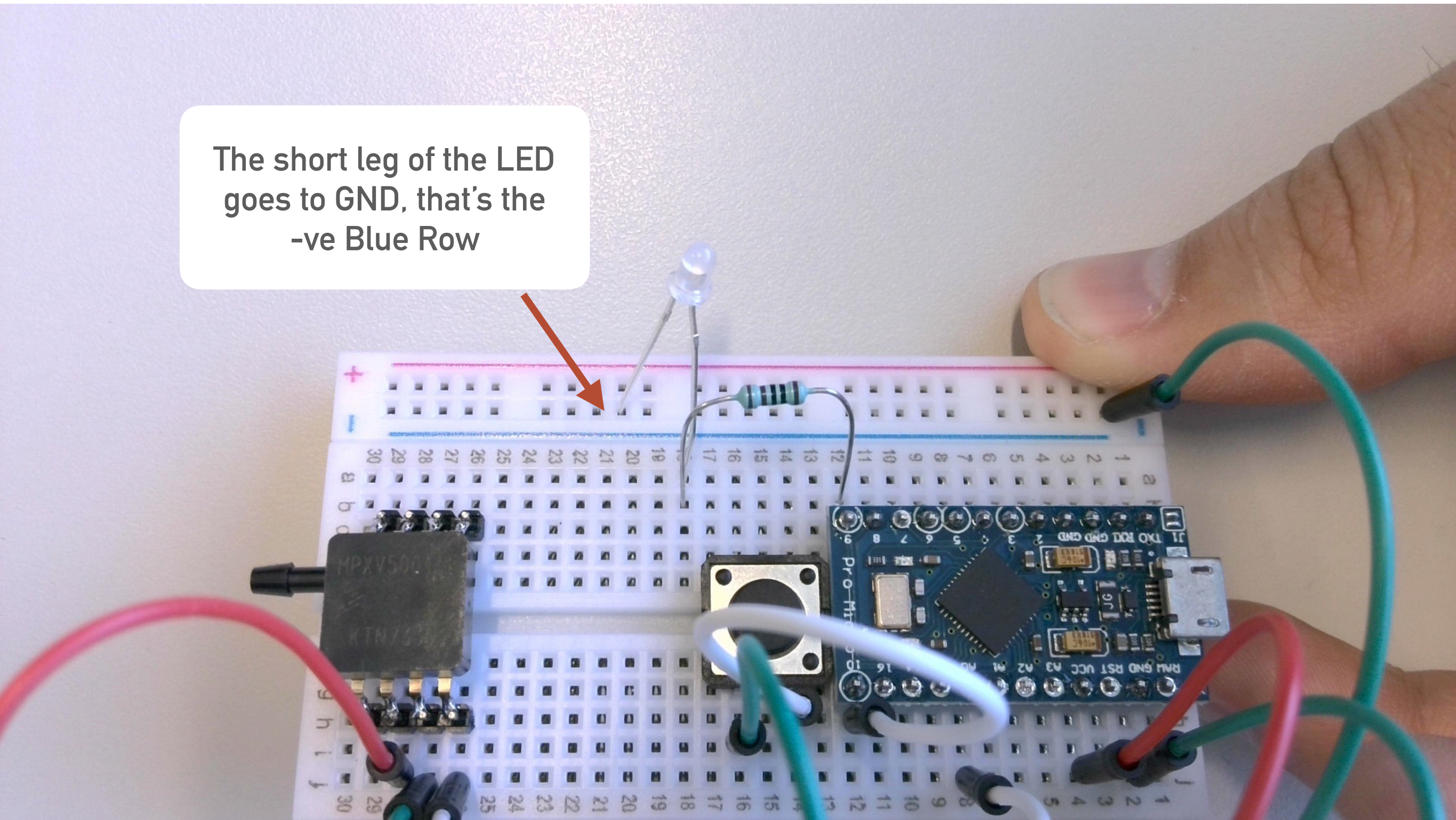
HOOKING UP YOUR SENSORS! BUTTON

.....



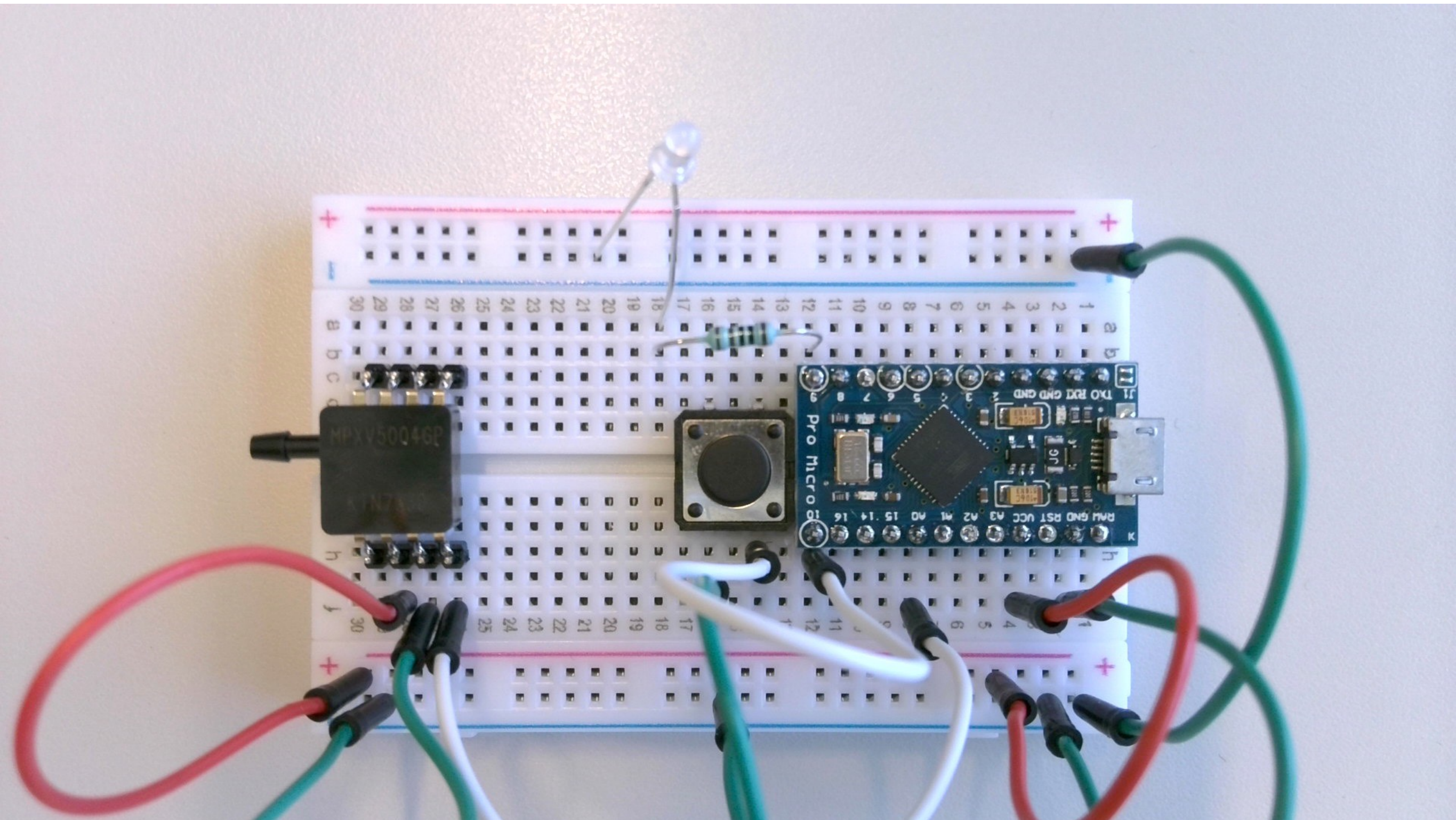
HOOKING UP YOUR SENSORS! LED

The short leg of the LED goes to GND, that's the -ve Blue Row



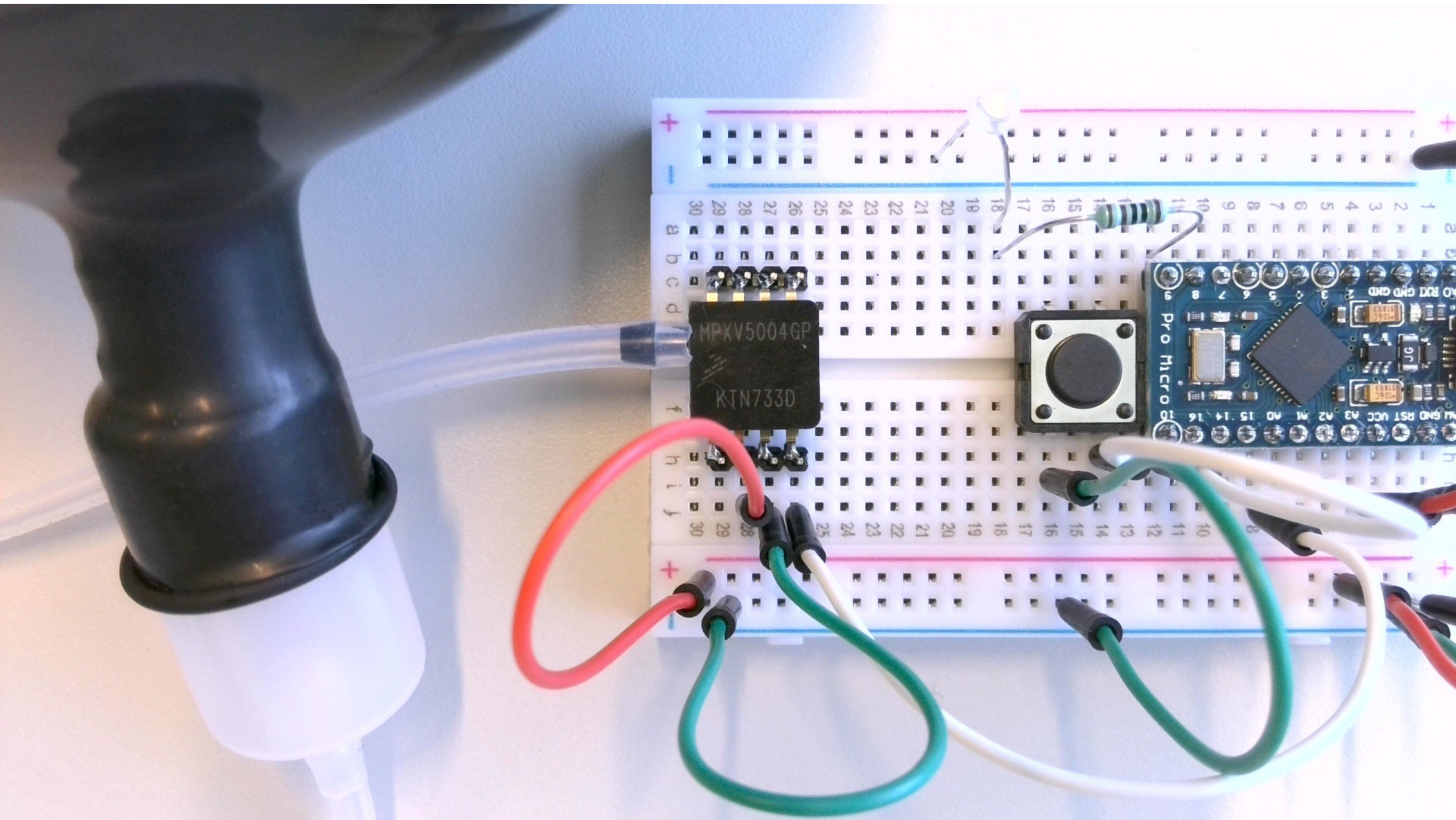
HOOKING UP YOUR SENSORS! LED

.....



HOOKING UP YOUR SENSORS! ATTACH A BALLOON

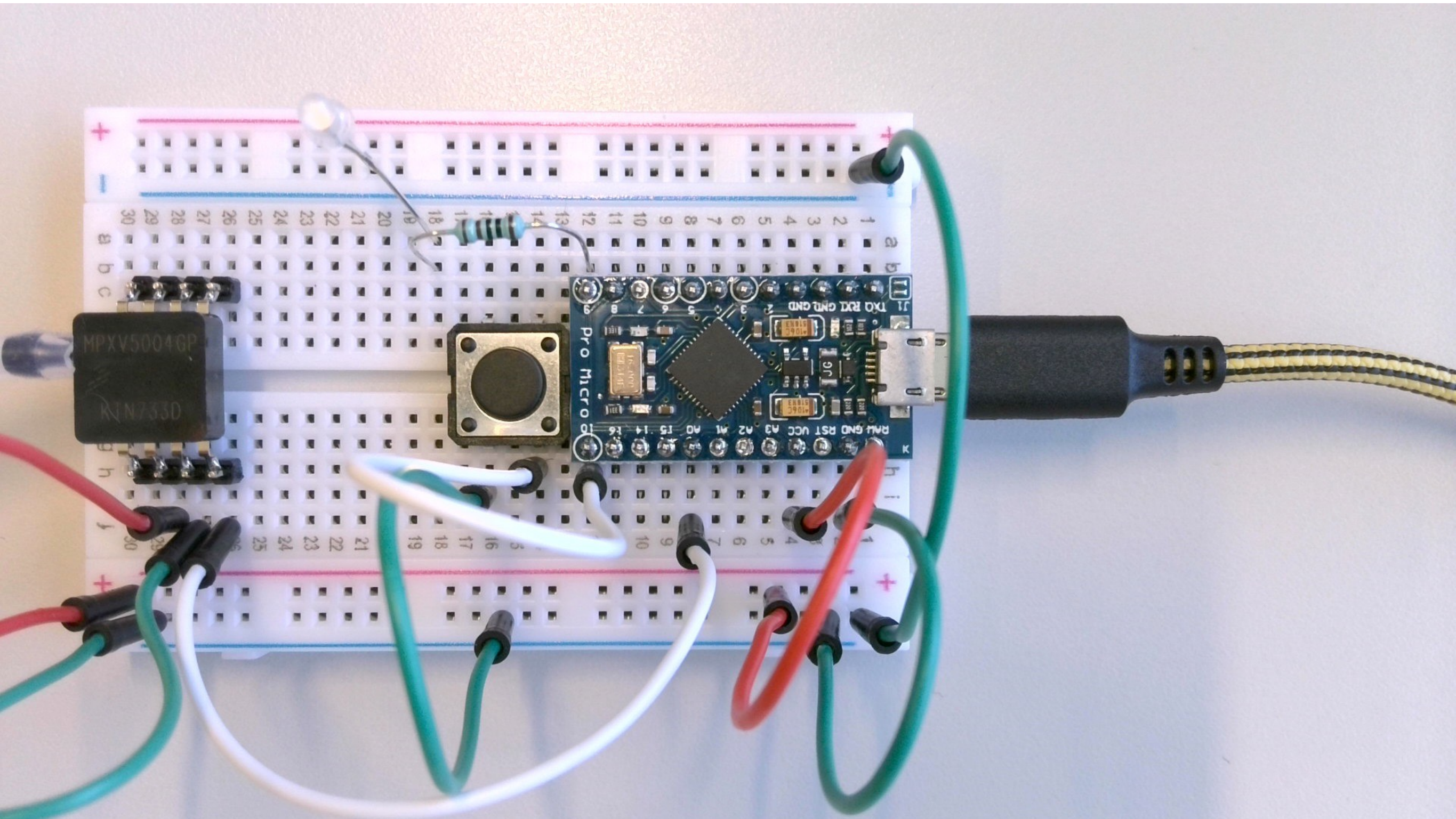
.....



SAFETY CHECK!

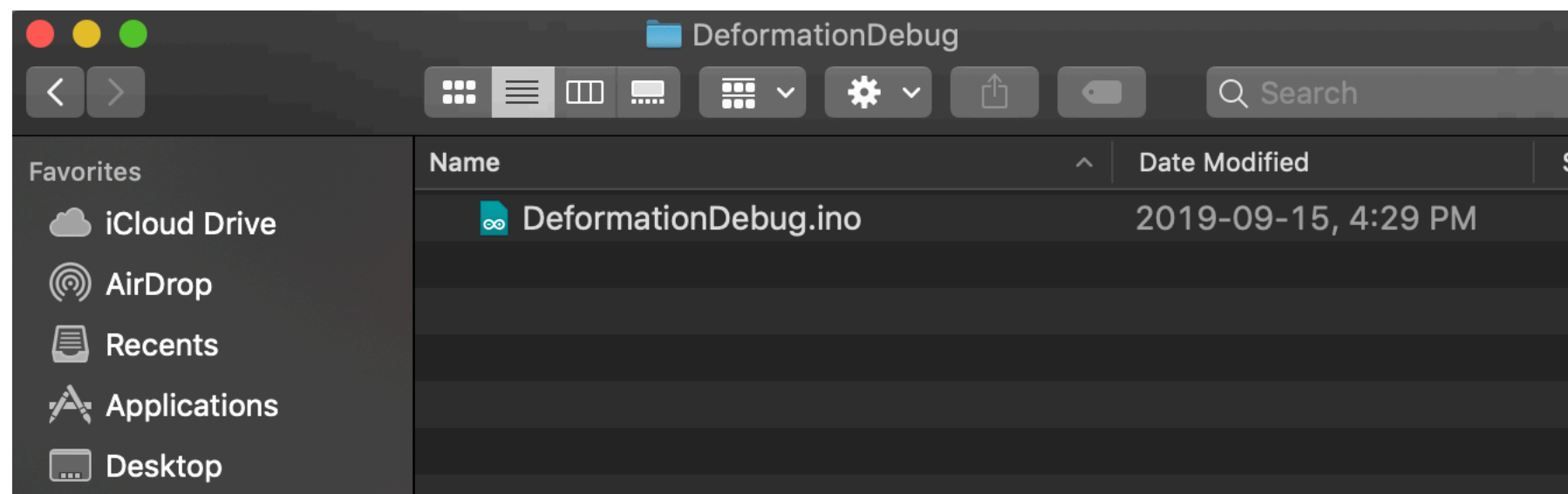
*BEFORE PLUGGING THE
USB CABLE INTO YOUR
COMPUTER GET MATT TO
CHECK YOUR BOARD SO
YOU KNOW IT'S SAFE AND
NOTHING WILL BREAK!*

HOOKING UP YOUR SENSORS! PLUG IN USB



ON YOUR COMPUTER

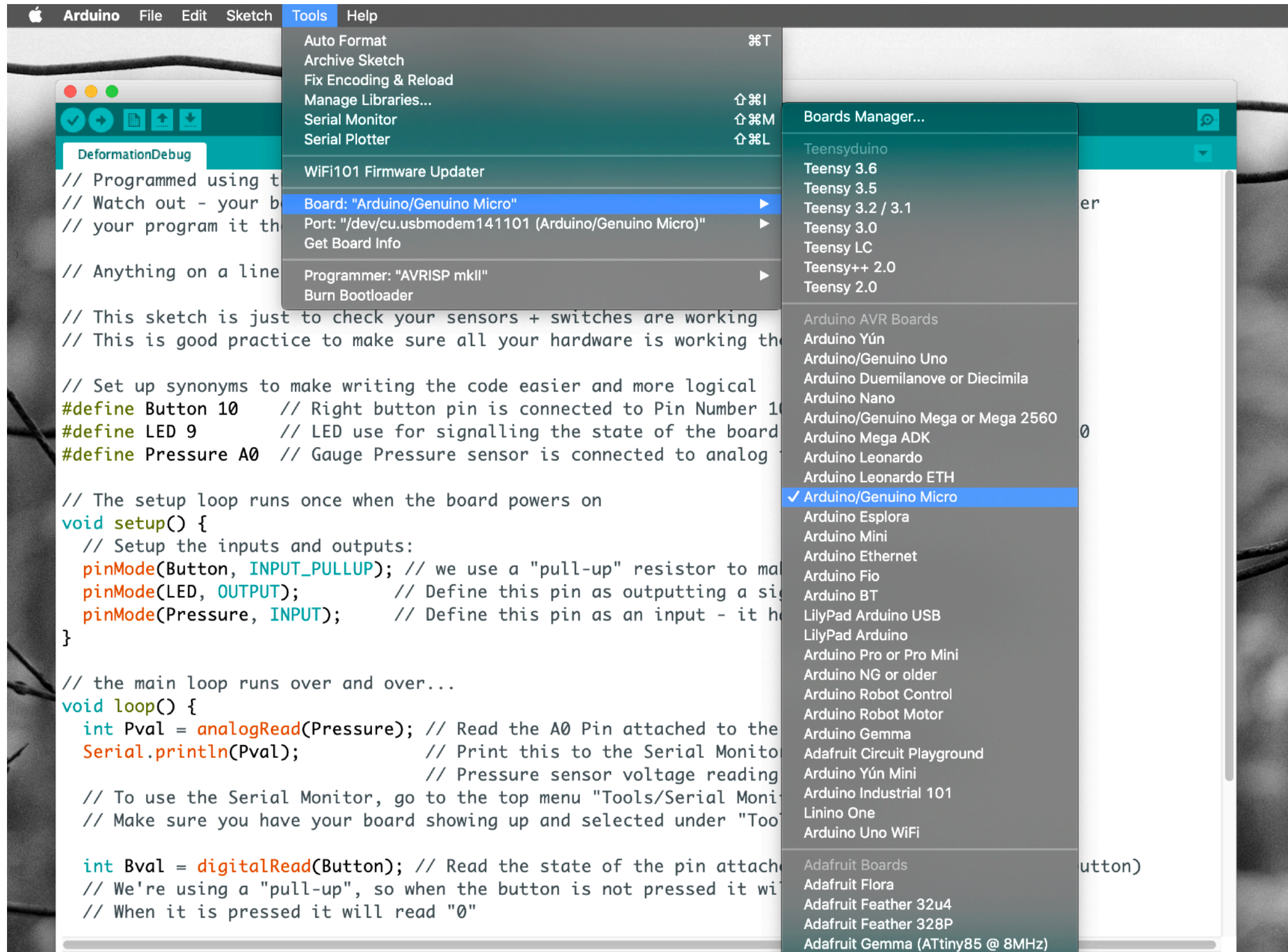
Launch Arduino and open “DeformationDebug.ino”



Arduino IDE - a software platform used to program your microcontroller.

<https://www.arduino.cc/en/Main/Software>

PROGRAM YOUR DEVICE – BOARD TYPE

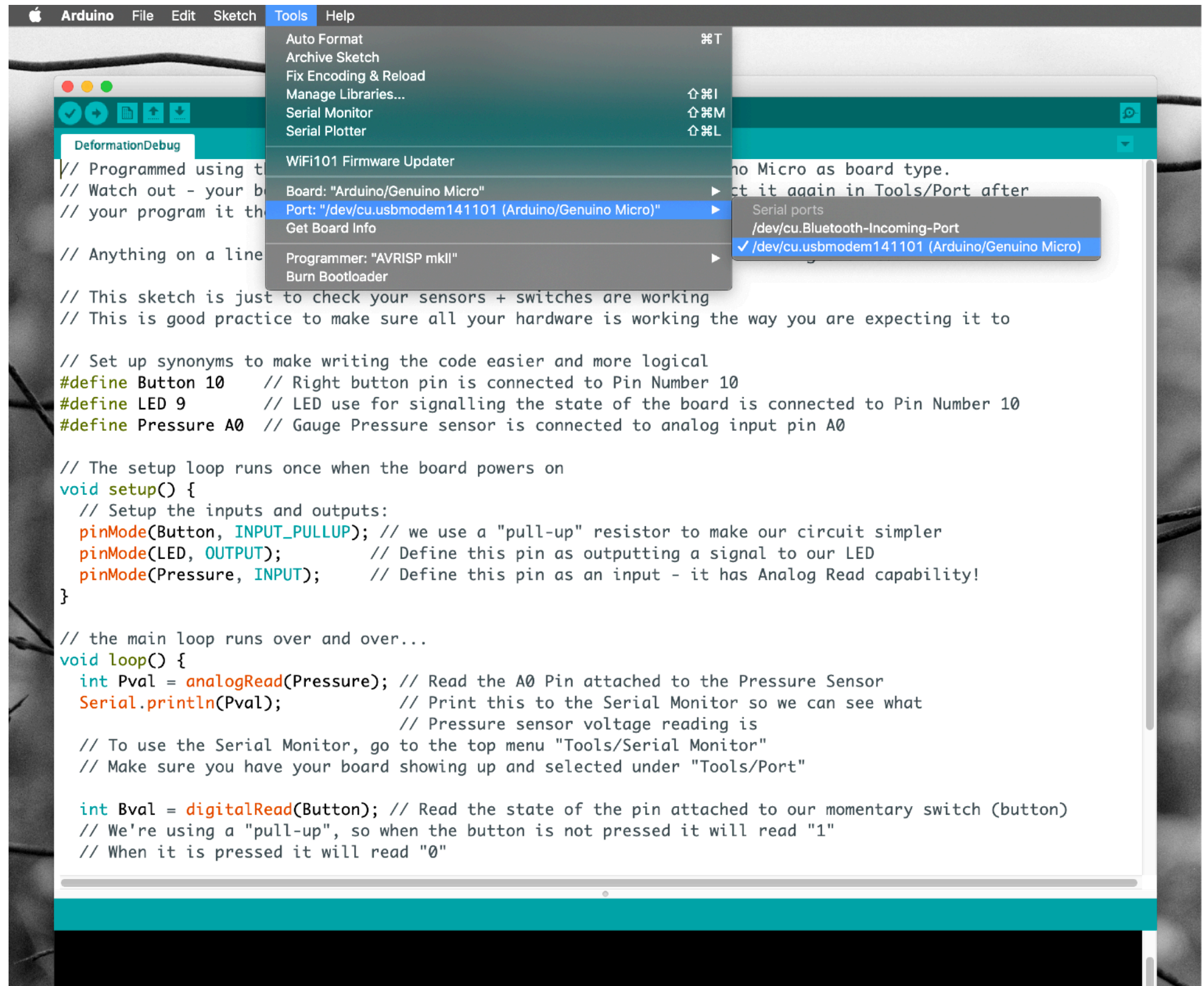


PROGRAM YOUR DEVICE – PORT

.....

Sometime this name changes after programming, so you may need to reselect this from time to time.

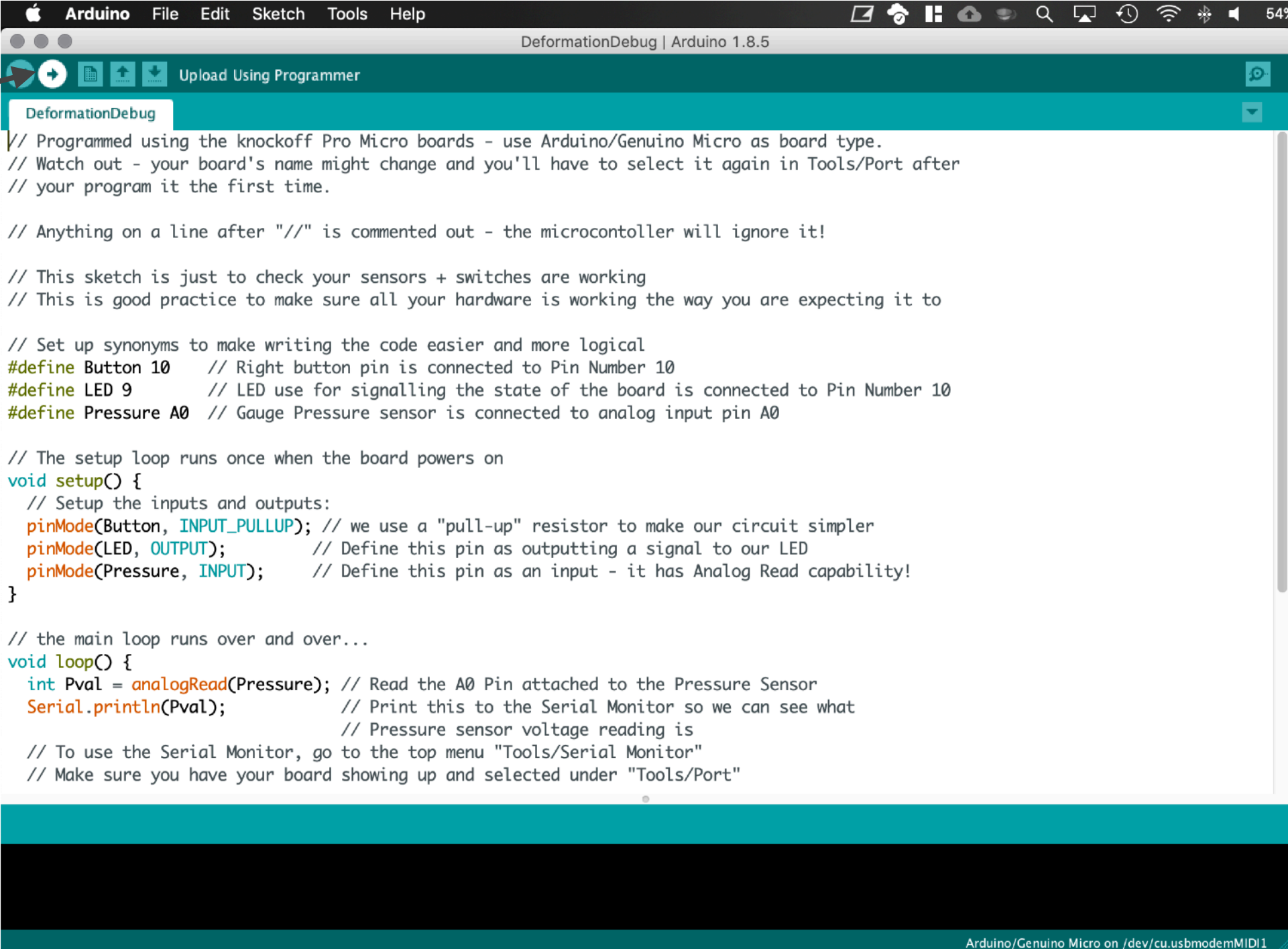
If you get an error that says “board not found” you should check it’s plugged in and that your port is selected correctly.



PROGRAM YOUR DEVICE – DEFORMATIONDEBUG.INO

.....

With your arduino plugged in, press the ARROW button to compile the code and send it to the arduino.



```
Arduino  File  Edit  Sketch  Tools  Help
DeformationDebug | Arduino 1.8.5

Upload Using Programmer

DeformationDebug
// Programmed using the knockoff Pro Micro boards - use Arduino/Genuino Micro as board type.
// Watch out - your board's name might change and you'll have to select it again in Tools/Port after
// your program it the first time.

// Anything on a line after "/" is commented out - the microcontroller will ignore it!

// This sketch is just to check your sensors + switches are working
// This is good practice to make sure all your hardware is working the way you are expecting it to

// Set up synonyms to make writing the code easier and more logical
#define Button 10    // Right button pin is connected to Pin Number 10
#define LED 9        // LED use for signalling the state of the board is connected to Pin Number 10
#define Pressure A0 // Gauge Pressure sensor is connected to analog input pin A0

// The setup loop runs once when the board powers on
void setup() {
  // Setup the inputs and outputs:
  pinMode(Button, INPUT_PULLUP); // we use a "pull-up" resistor to make our circuit simpler
  pinMode(LED, OUTPUT);          // Define this pin as outputting a signal to our LED
  pinMode(Pressure, INPUT);      // Define this pin as an input - it has Analog Read capability!
}

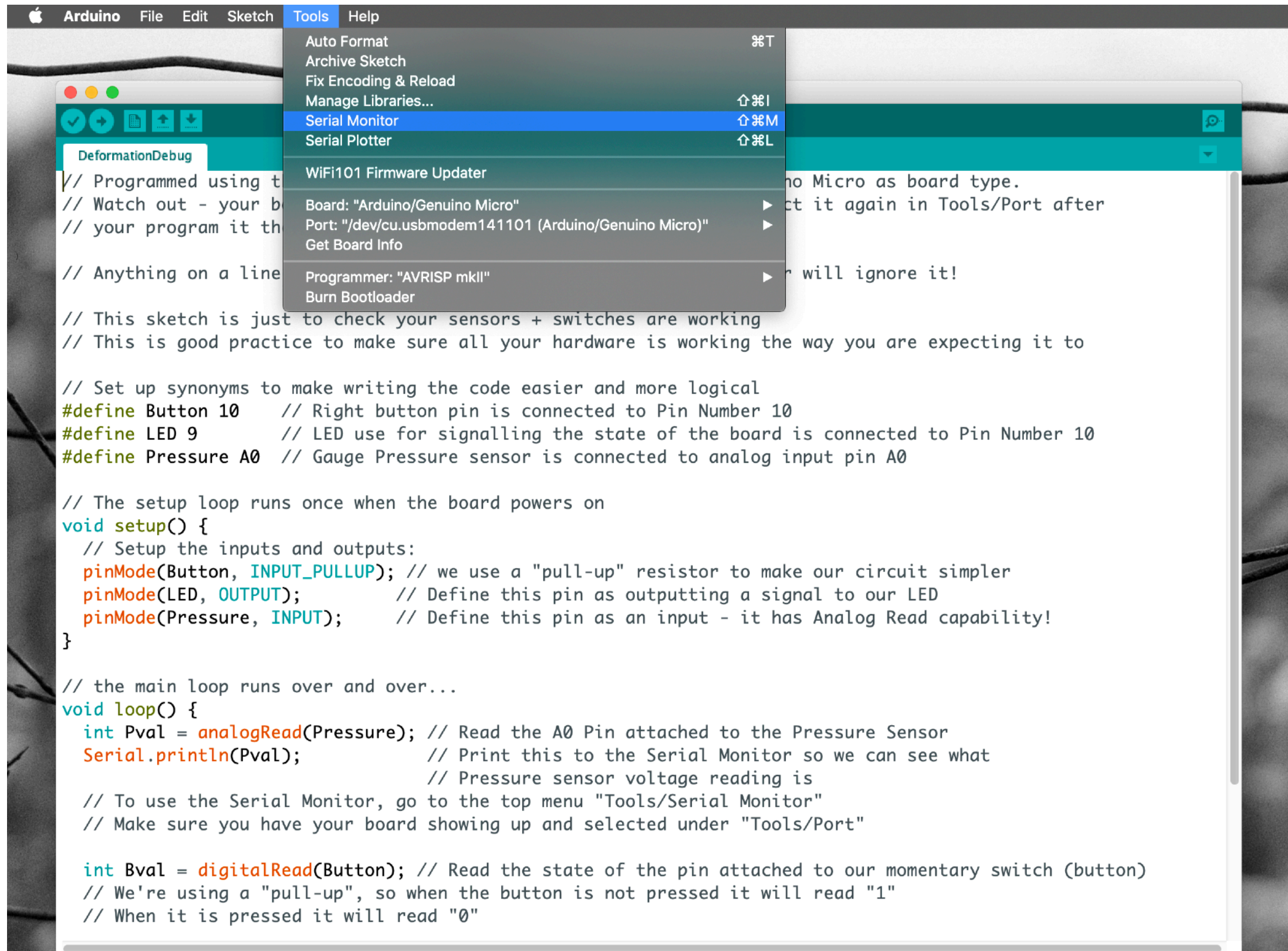
// the main loop runs over and over...
void loop() {
  int Pval = analogRead(Pressure); // Read the A0 Pin attached to the Pressure Sensor
  Serial.println(Pval);            // Print this to the Serial Monitor so we can see what
                                   // Pressure sensor voltage reading is

  // To use the Serial Monitor, go to the top menu "Tools/Serial Monitor"
  // Make sure you have your board showing up and selected under "Tools/Port"

Arduino/Genuino Micro on /dev/cu.usbmodemMIDI1
```


PROGRAM YOUR DEVICE – MONITOR THE SERIAL PORT

.....

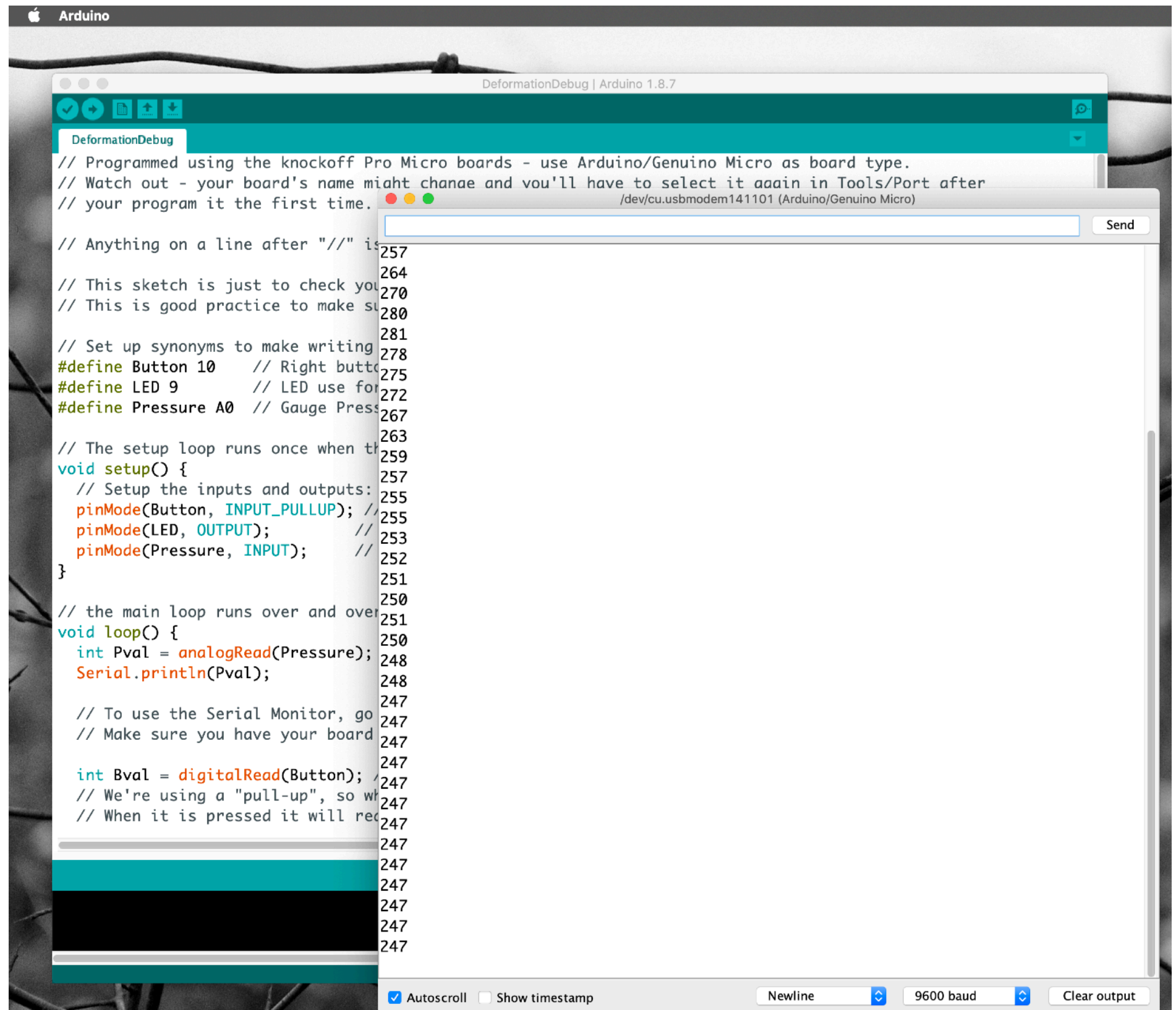


PROGRAM YOUR DEVICE – CHECK PRESSURE READING

.....

Squeeze your balloon and see how the values change!

This is printing out the analog voltage output by the sensor and converted to a number between 0-1023.

The image shows a screenshot of the Arduino IDE interface. The main window displays a C++ sketch titled "DeformationDebug" for an Arduino 1.8.7. The sketch includes comments about using knockoff Pro Micro boards and a code structure with a setup loop and a main loop. The setup loop initializes a button (pin 10), an LED (pin 9), and a pressure sensor (pin A0). The main loop reads the pressure sensor's analog value and prints it to the serial monitor. The serial monitor is open, showing a series of "247" values, indicating the current pressure reading. The IDE's status bar at the bottom shows "Autoscroll" is checked, "Show timestamp" is unchecked, and the serial port is set to "9600 baud".

```
DeformationDebug | Arduino 1.8.7
// Programmed using the knockoff Pro Micro boards - use Arduino/Genuino Micro as board type.
// Watch out - your board's name might change and you'll have to select it again in Tools/Port after
// your program it the first time.
// Anything on a line after "//" is
// This sketch is just to check you
// This is good practice to make sure
// Set up synonyms to make writing
#define Button 10 // Right button
#define LED 9 // LED use for
#define Pressure A0 // Gauge Pressure

// The setup loop runs once when the board is first powered up
void setup() {
  // Setup the inputs and outputs:
  pinMode(Button, INPUT_PULLUP);
  pinMode(LED, OUTPUT);
  pinMode(Pressure, INPUT);
}

// the main loop runs over and over again
void loop() {
  int Pval = analogRead(Pressure);
  Serial.println(Pval);

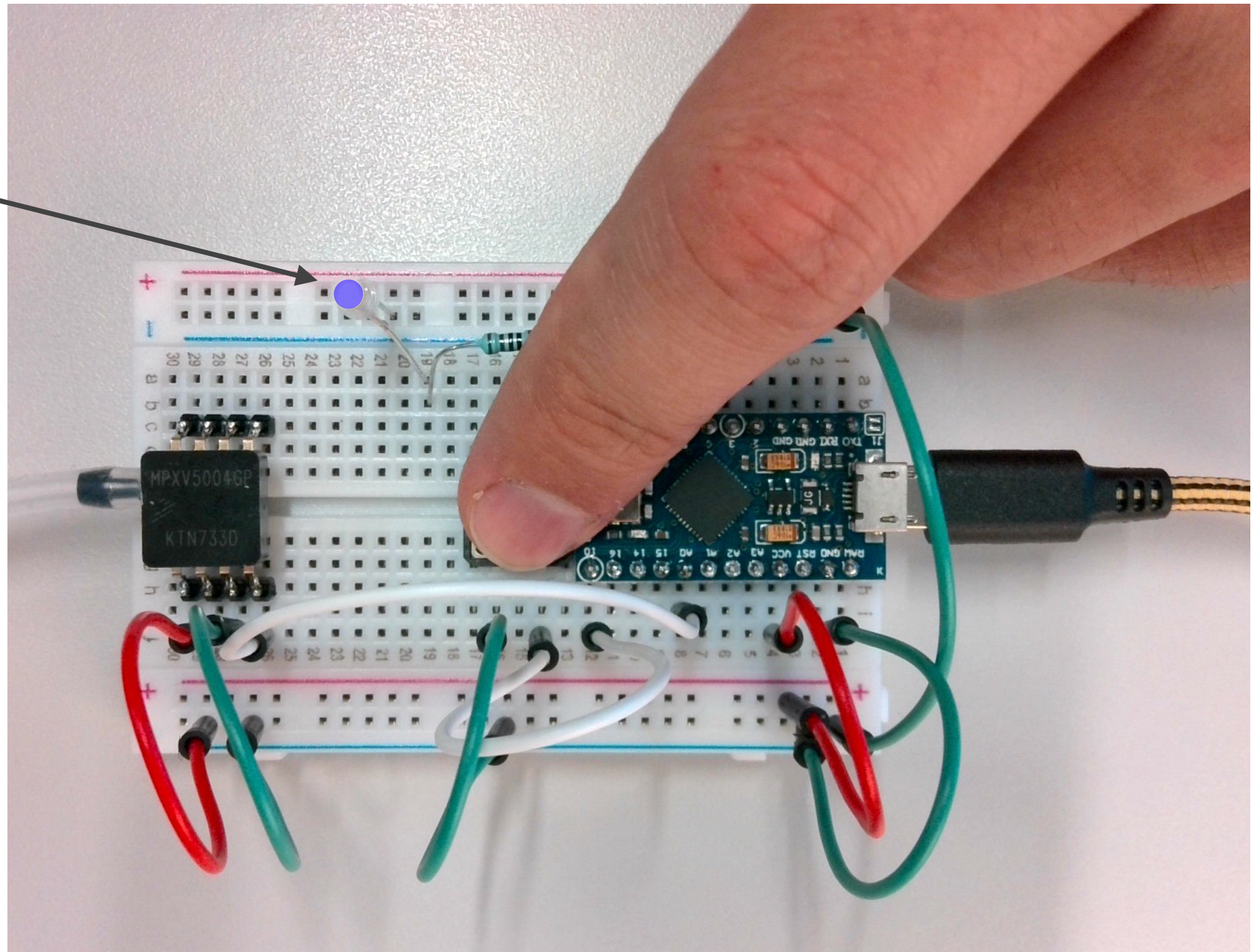
  // To use the Serial Monitor, go to Tools -> Serial Monitor
  // Make sure you have your board selected in the Tools menu

  int Bval = digitalRead(Button);
  // We're using a "pull-up", so when the button is pressed the value will be 0
  // When it is pressed it will read 0
}
```


PROGRAM YOUR DEVICE – CHECK BUTTON AND LED

Without the button pressed the LED should be bright blue.

When you press the button the LED should turn off.



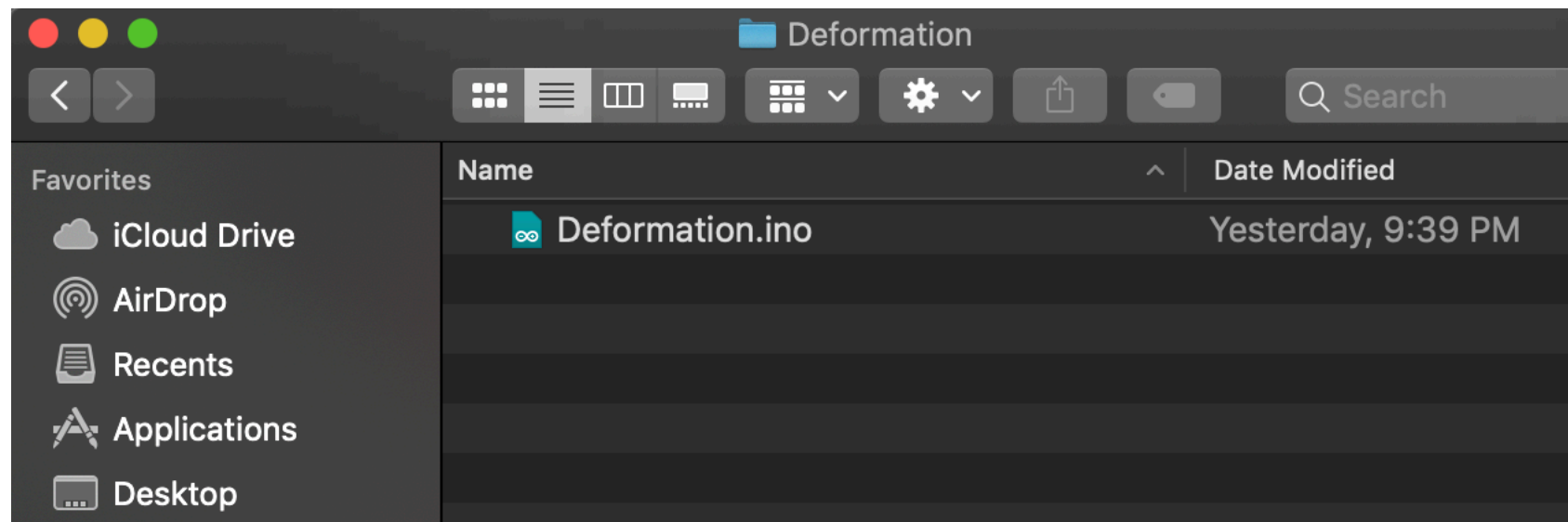
PROGRAM YOUR DEVICE – DEFORMATIONDEBUG.INO

If your button changed the LED from on to off, and your serial monitor values changed when you squeezed your balloon, you're good to go!

If it didn't, get some help to figure out what's wrong with how you hooked things up on your breadboard!

OPEN A FILE

Launch Arduino and open “Deformation.ino”

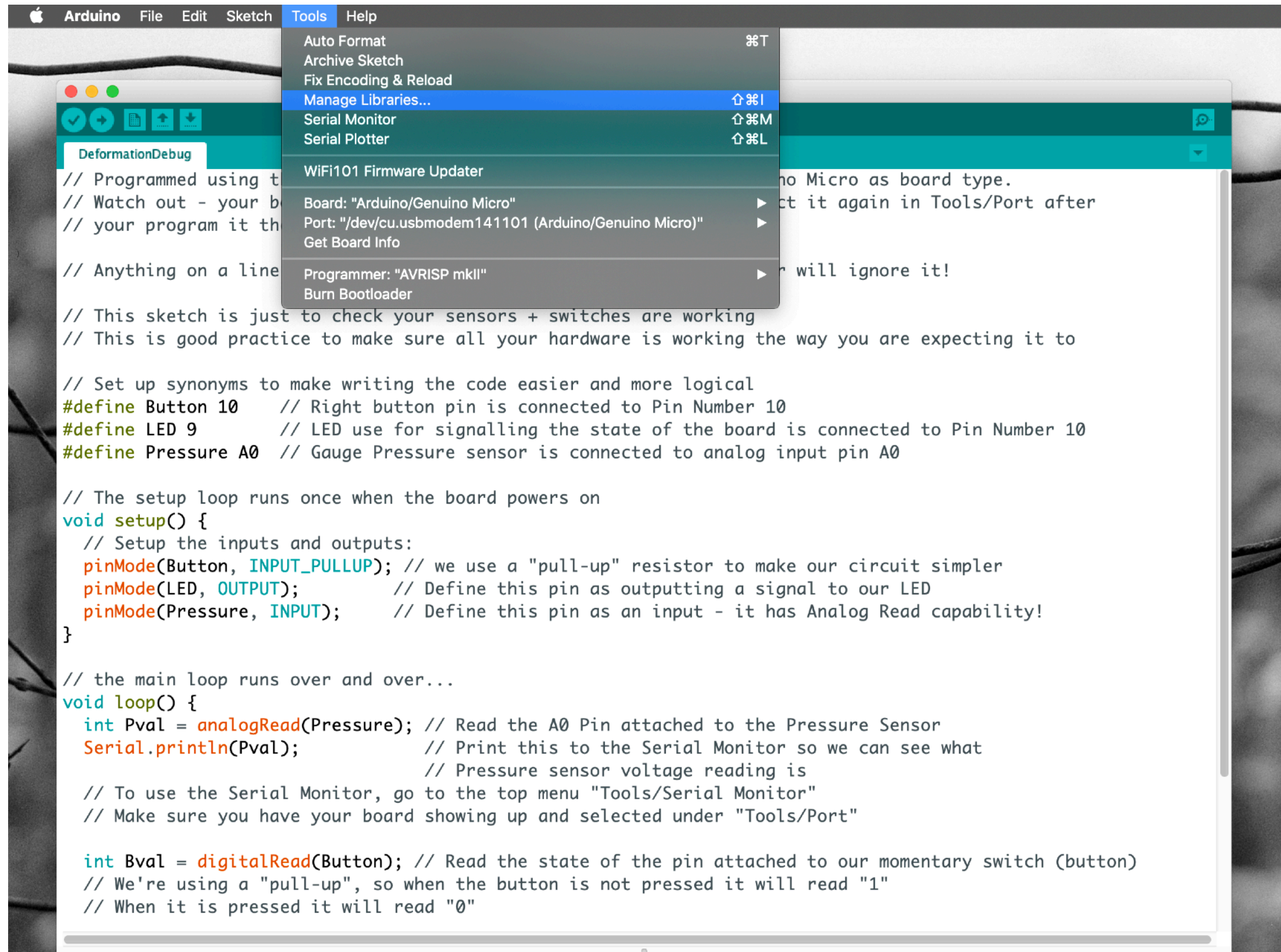


Arduino IDE - a software platform used to program your microcontroller.

<https://www.arduino.cc/en/Main/Software>

LIBRARIES – INSTALL THESE TO ADD FUNCTIONS TO YOUR PROGRAM

.....

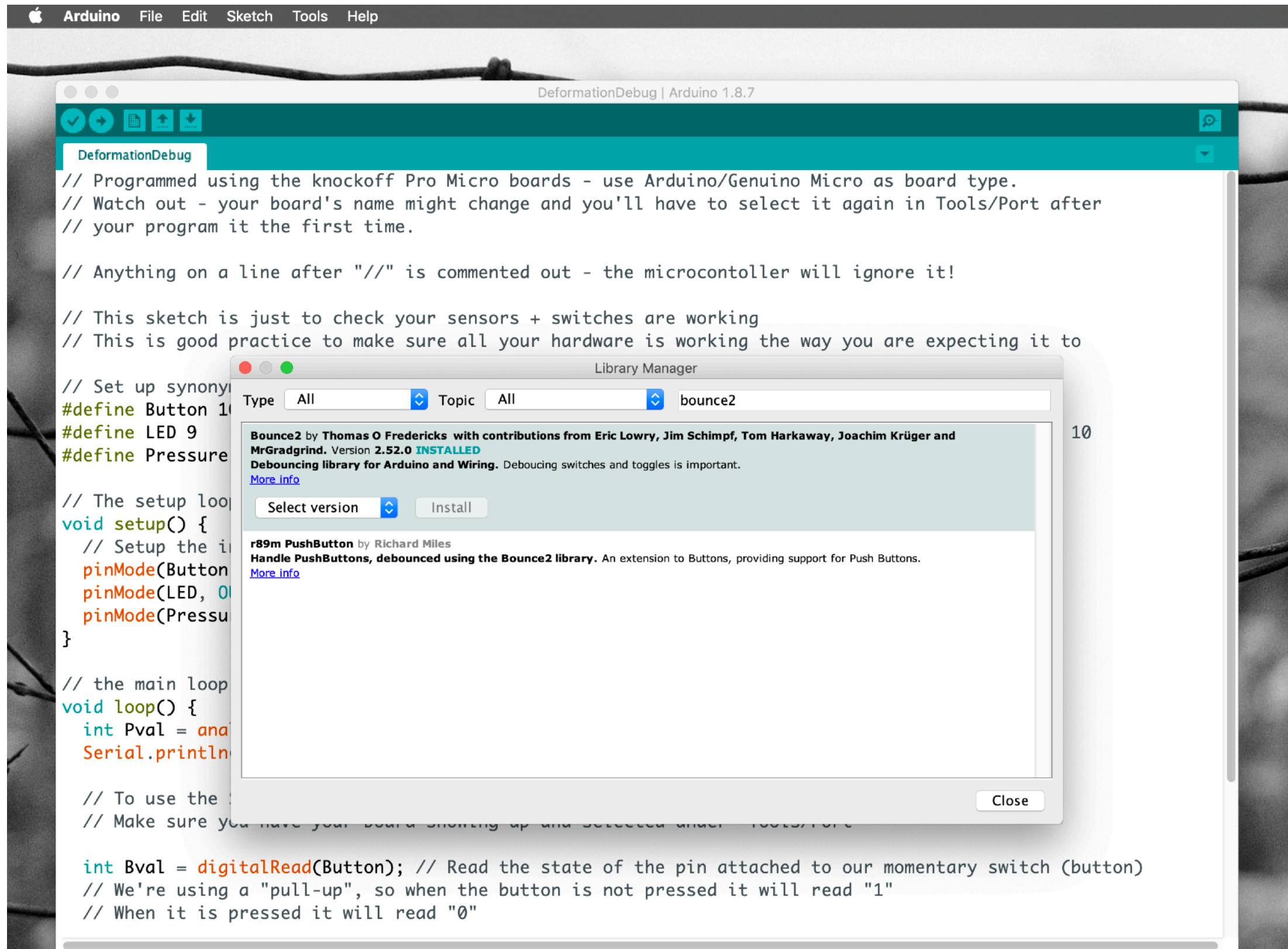


LIBRARIES – ELAPSEDMILLIS

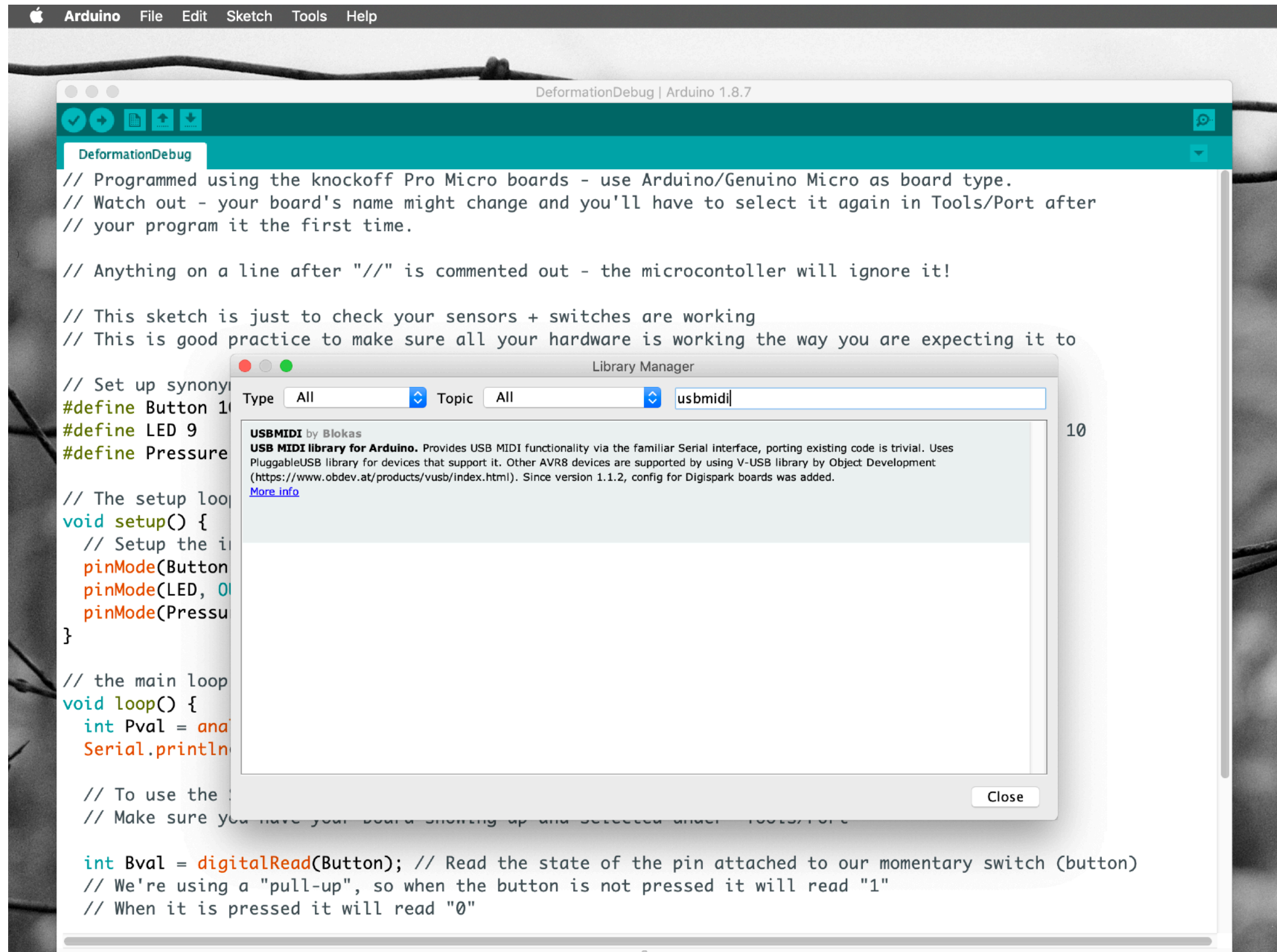


LIBRARIES – BOUNCE2

.....



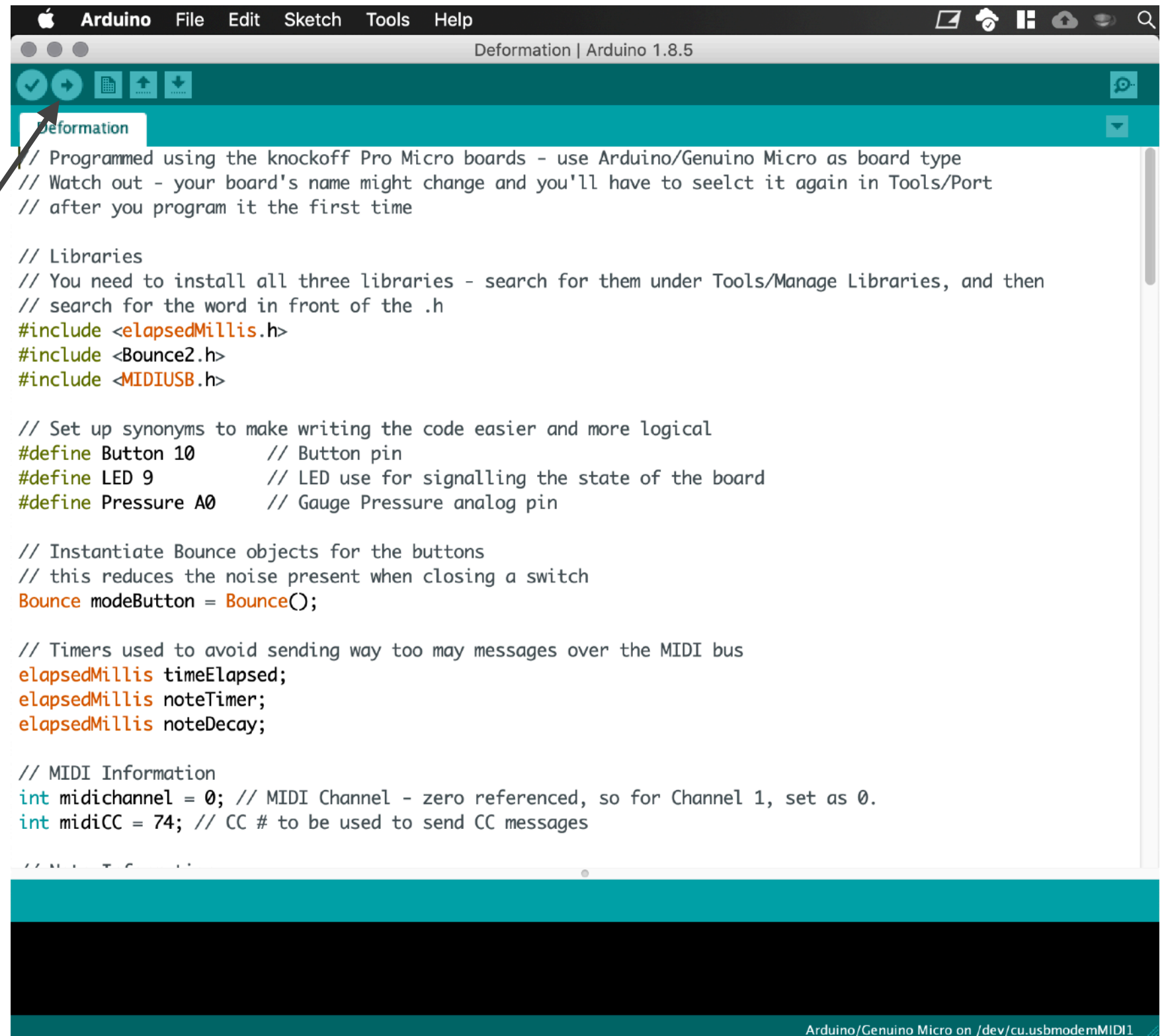
LIBRARIES – USBMIDI



PROGRAM YOUR DEVICE – DEFORMATION.INO

.....

Now program your board with the actual code. This will send out MIDI messages to your computer that HELM will turn into sounds.



```
Arduino  File  Edit  Sketch  Tools  Help
Deformation | Arduino 1.8.5

// Programmed using the knockoff Pro Micro boards - use Arduino/Genuino Micro as board type
// Watch out - your board's name might change and you'll have to select it again in Tools/Port
// after you program it the first time

// Libraries
// You need to install all three libraries - search for them under Tools/Manage Libraries, and then
// search for the word in front of the .h
#include <elapsedMillis.h>
#include <Bounce2.h>
#include <MIDIUSB.h>

// Set up synonyms to make writing the code easier and more logical
#define Button 10      // Button pin
#define LED 9          // LED use for signalling the state of the board
#define Pressure A0    // Gauge Pressure analog pin

// Instantiate Bounce objects for the buttons
// this reduces the noise present when closing a switch
Bounce modeButton = Bounce();

// Timers used to avoid sending way too many messages over the MIDI bus
elapsedMillis timeElapsed;
elapsedMillis noteTimer;
elapsedMillis noteDecay;

// MIDI Information
int midichannel = 0; // MIDI Channel - zero referenced, so for Channel 1, set as 0.
int midiCC = 74; // CC # to be used to send CC messages

// ...
```

Arduino/Genuino Micro on /dev/cu.usbmodemMIDI1

PROGRAM YOUR DEVICE – MODES

Pressing your button will cycle through the modes:

MODE 0 - low LED

Send MIDI Notes

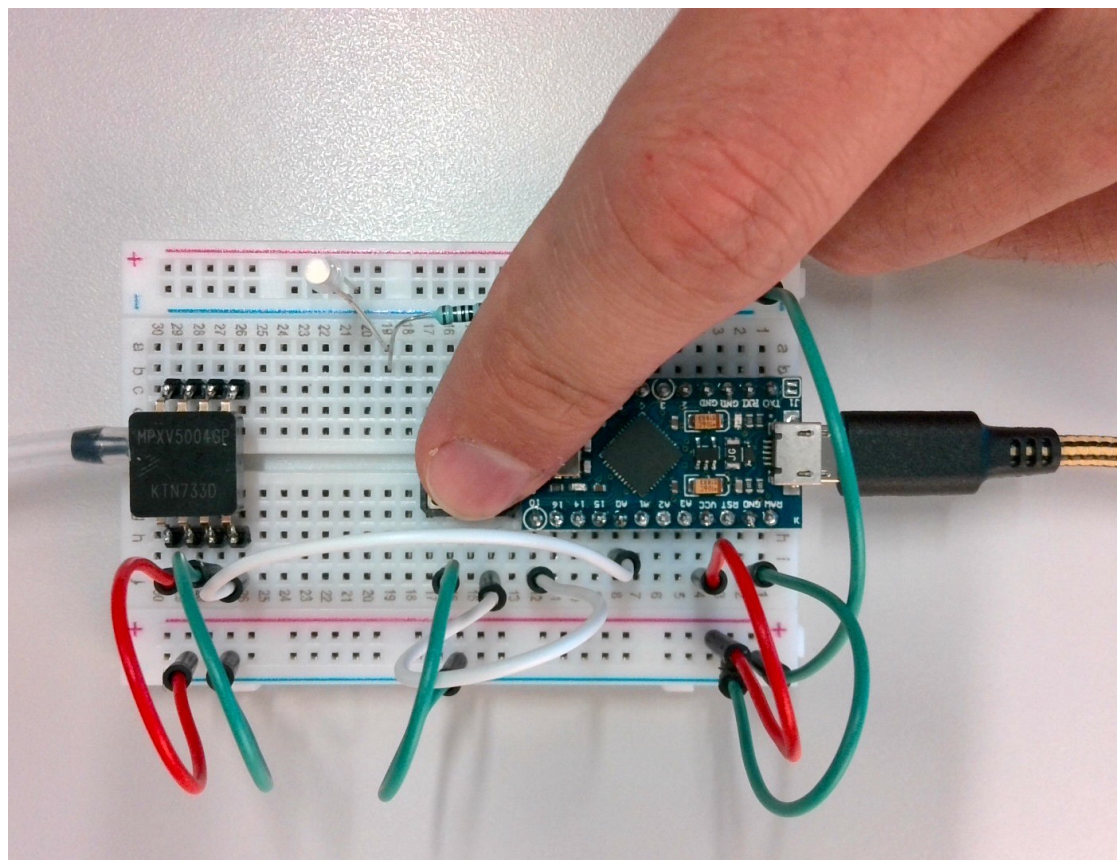
Send MIDI CC

MODE 1 - med LED

Send MIDI CC

MODE 2 - high LED

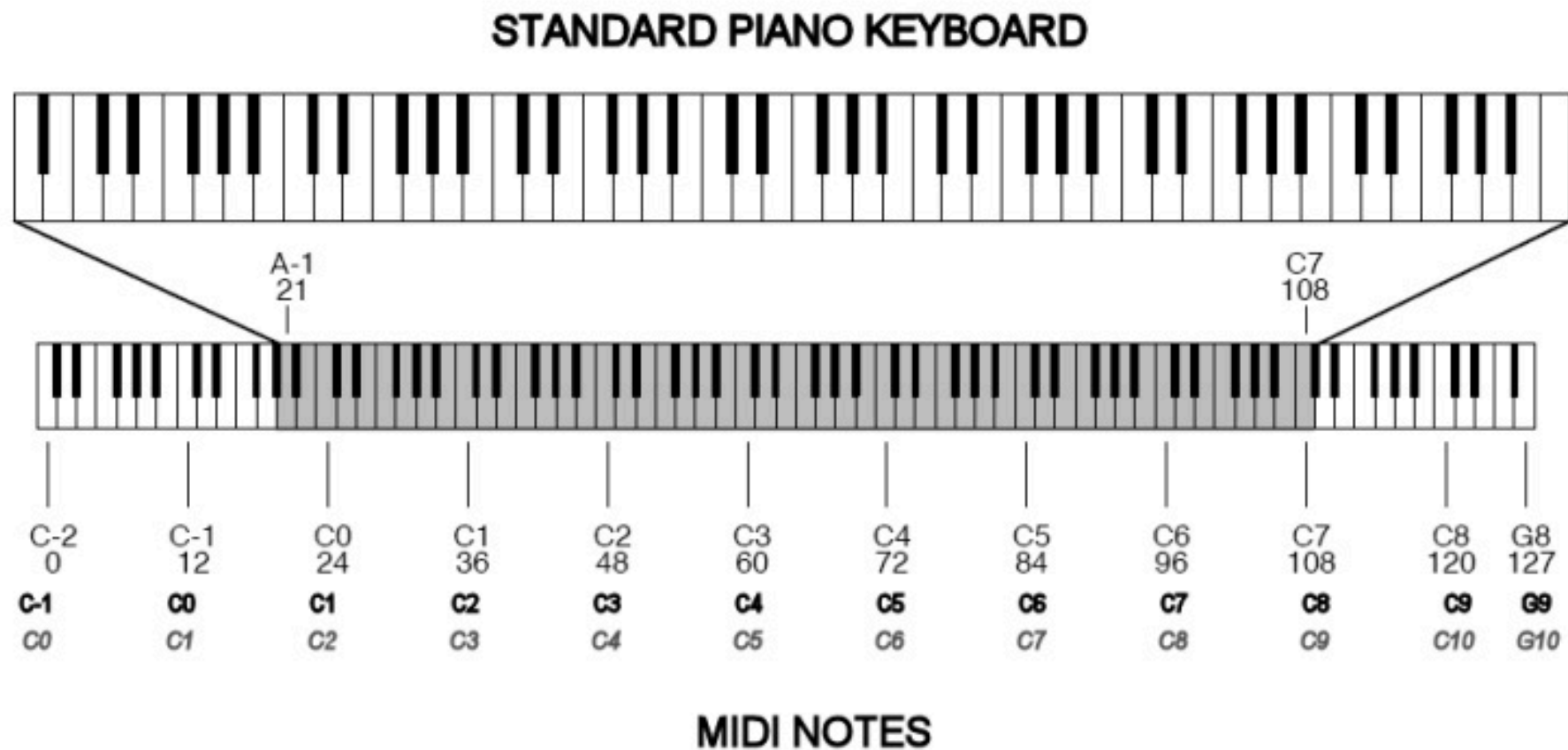
Set the High Pressure value - make sure you squeeze the balloon in this mode so the pressure range is set properly.



CHANGE YOUR NOTES!

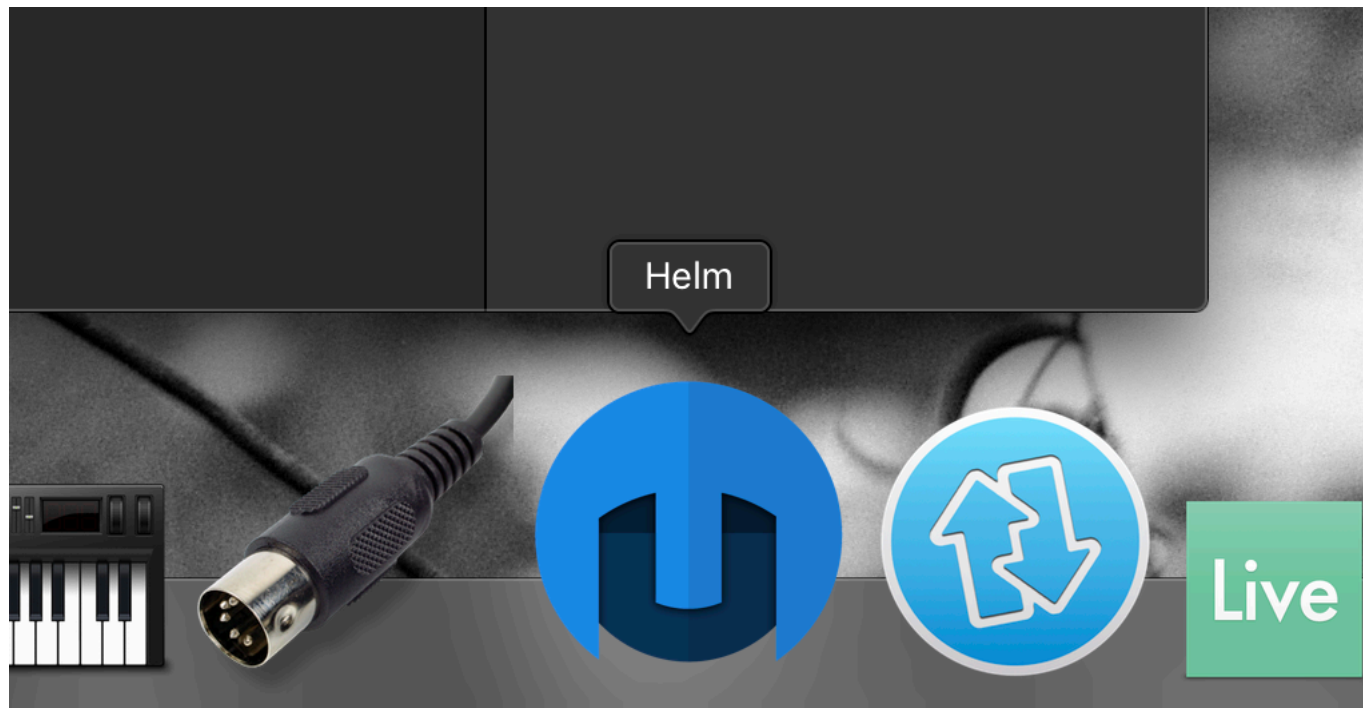
Each squeeze will trigger a note randomly from a set of pre-selected notes. You can change what those notes are and how many are in the set. If you make any changes you need to reprogram your arduino with the updated code.

```
// Note Information
// Pitch defines notes using MIDI note numbers - 60 is middle C, valid pitches are 0-127
// 0 is super low, and 127 is super high! Be wise about the pitches you include!
byte pitch[] = {60, 64, 67, 71, 72}; // You can change which pitch plays by changing these MIDI Note Numbers
int numNotes = 5; // this variable MUST match the number of notes in the pitch array in the previous line
```



<https://en.wikipedia.org/wiki/MIDI>

OPEN HELM – SOFTWARE SYNTHESIZER



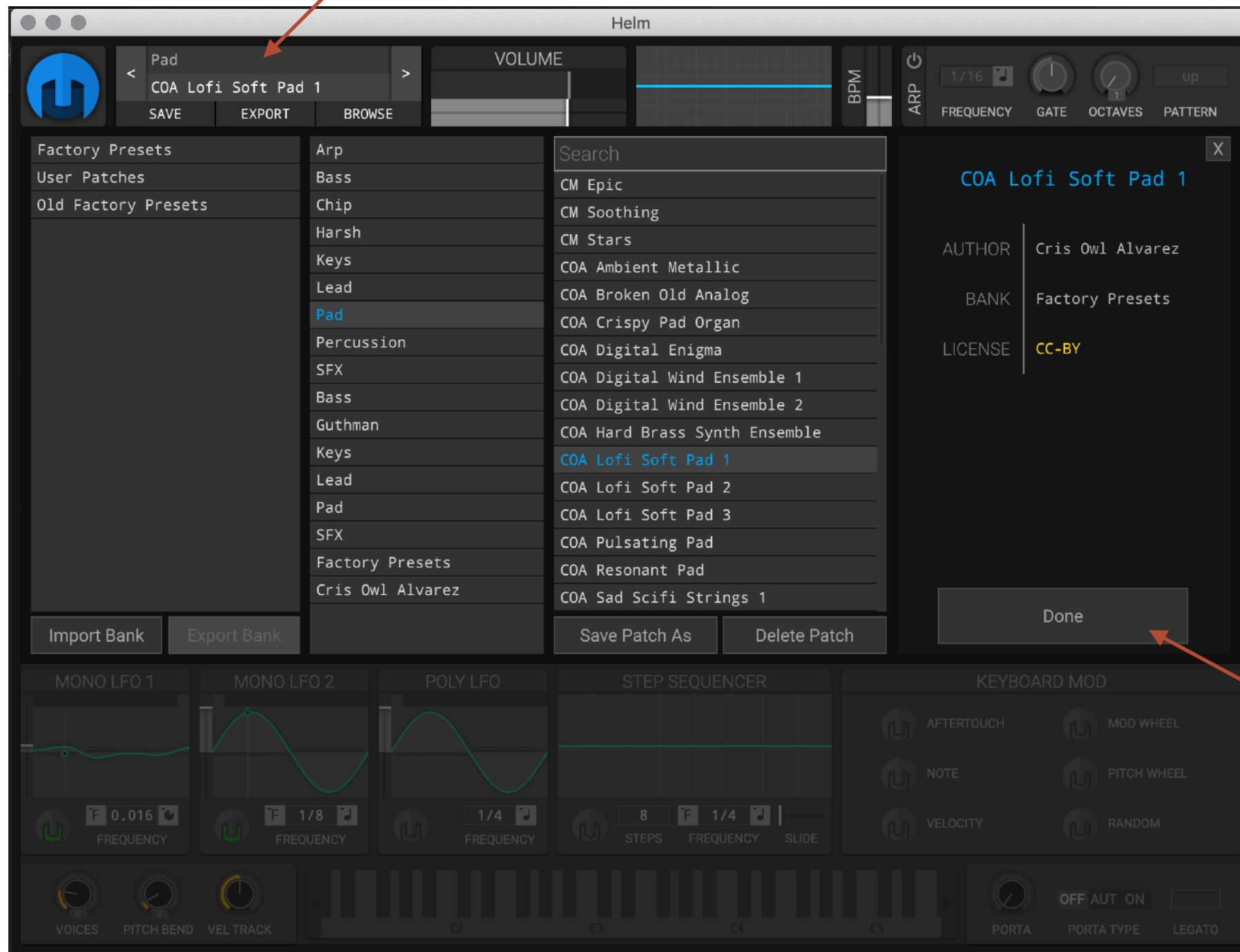
*Helm - a software synthesizer
to make musical sounds with
your computer.*

<https://tytel.org/helm/>

SETUP HELM

Click here to open the menu to select a sound

Make sure your laptop's speakers are on and turned up!



Squeeze your balloon quickly to send notes to Helm!

Try different sounds, then click done.

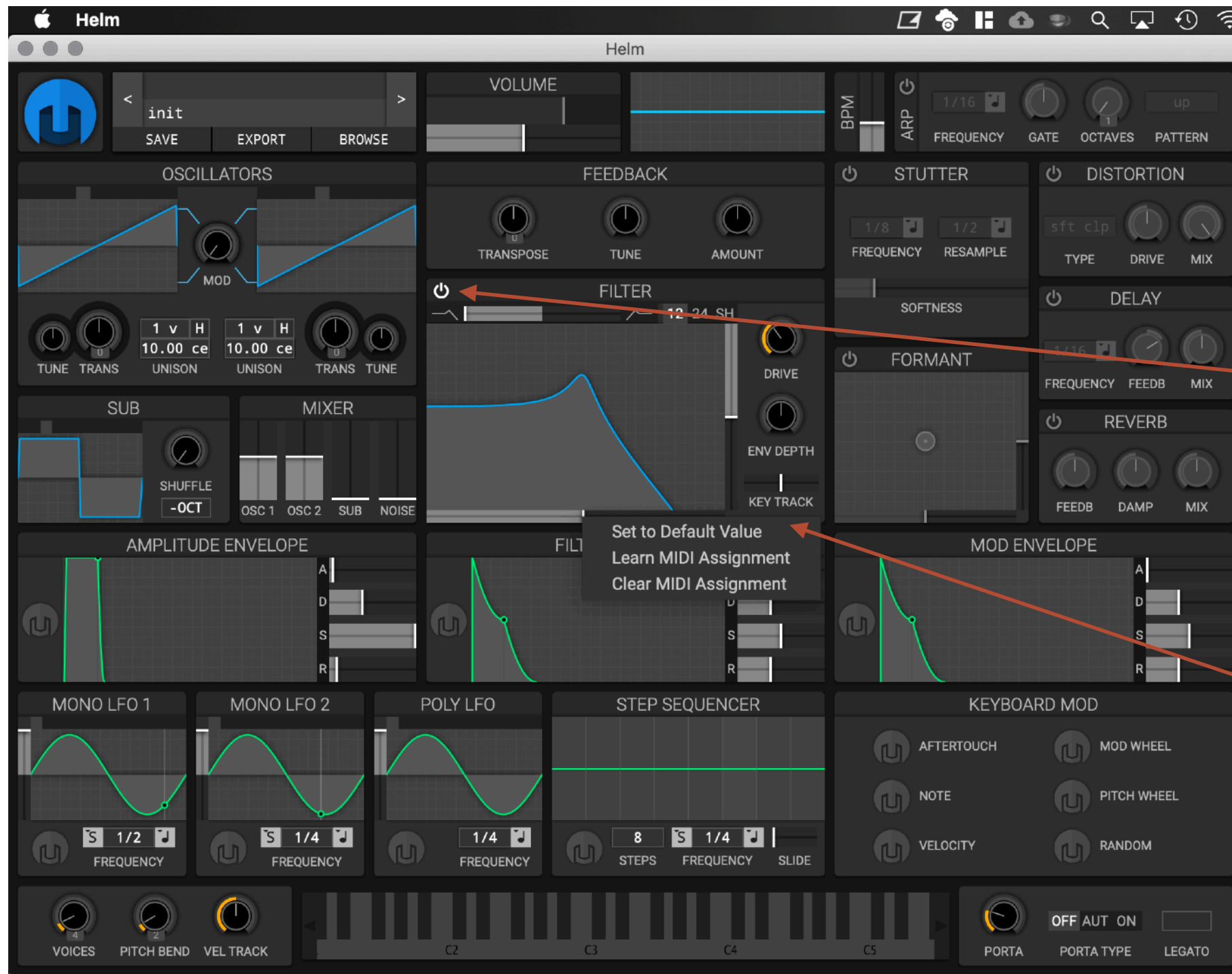
SETUP HELM



Make sure your laptop's speakers are on and turned up!

Squeeze your balloon quickly to send notes to Helm!

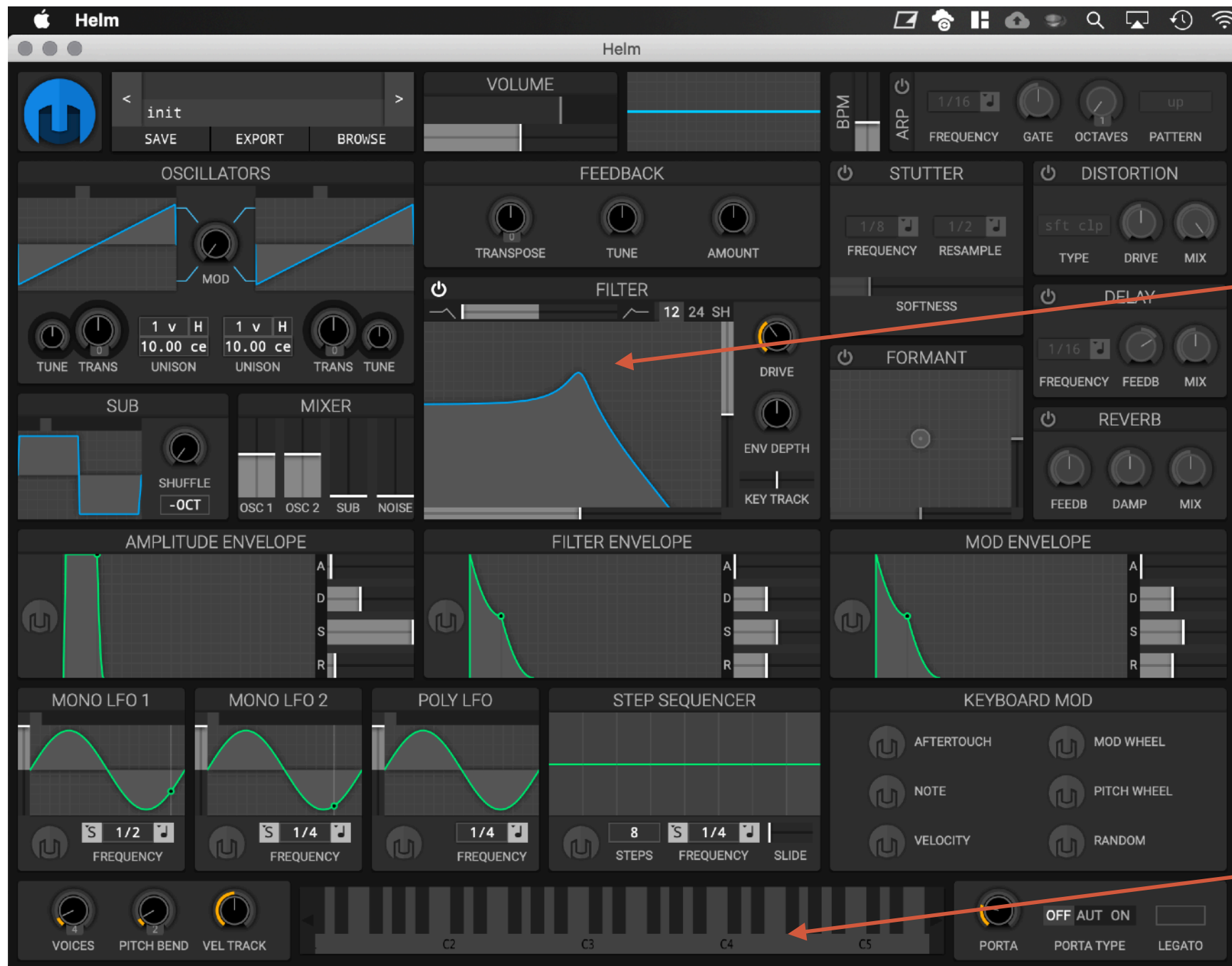
SETUP HELM



Make sure the filter is on - it shouldn't be greyed out. If it is greyed out, click the ON button here.

Right Click the bar below the filter, then click "Learn MIDI Assignment"

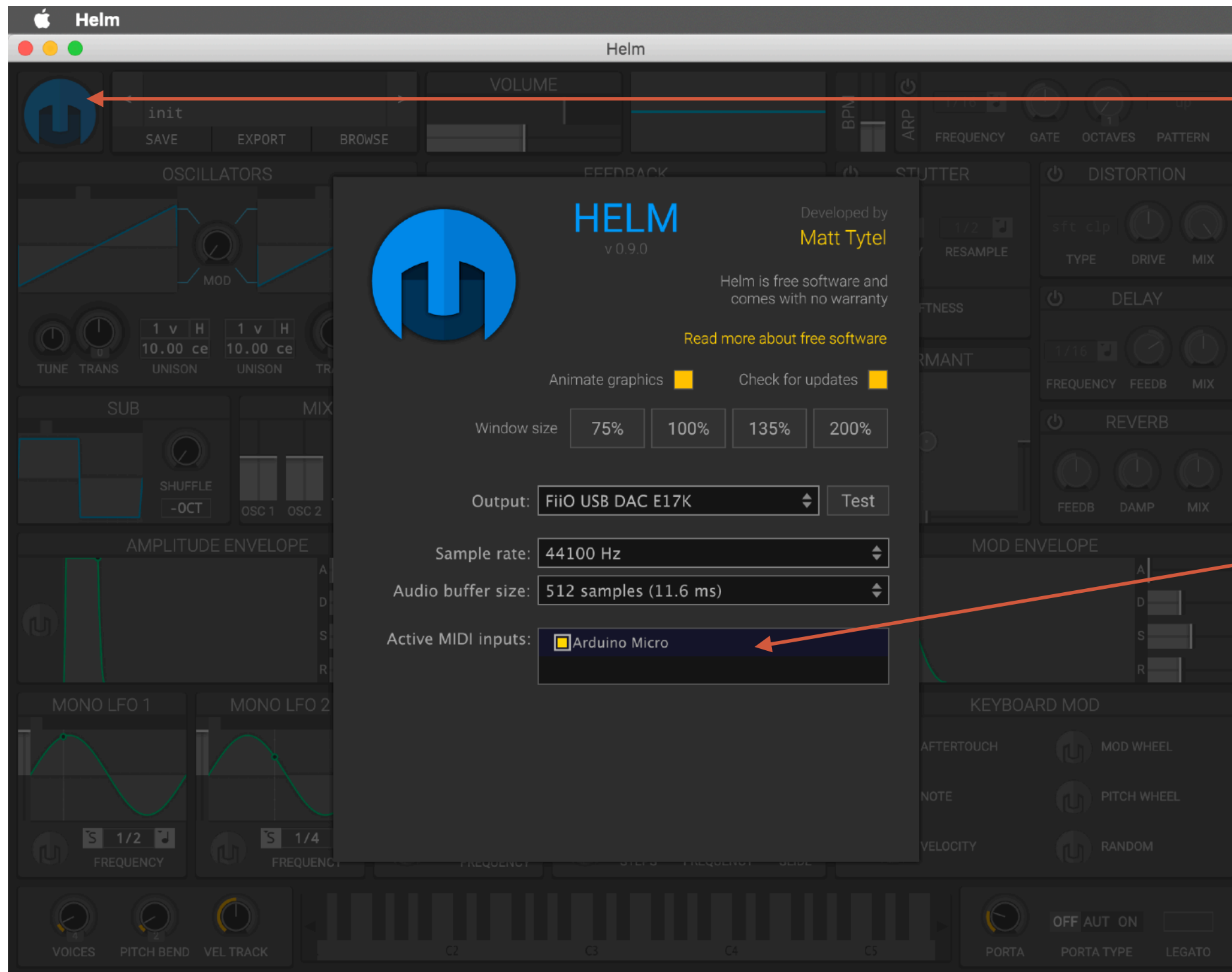
SETUP HELM



You should see the filter peak move back and forth as you squeeze your balloon!

You should see the keys of the piano lighting up as you squeeze your balloon!

NOT WORKING? CHECK THE MIDI PORT



Click the HELMet to get this dialog to open.

You should see “Arduino Micro” as a MIDI input, with a yellow box showing it is selected.