



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China



□

UNTZtrument: a Trellis MIDI Instrument

Created by Phillip Burgess



Last updated on 2016-12-06 07:47:26 PM UTC

Guide Contents

Guide Contents	2
Overview	3
What UNTZtrument is:	4
What UNTZtrument is not:	6
First Steps	8
Let's Get Started!	9
Introducing Adafruit Trellis (http://adafru.it/dxx)	14
Troubleshooting	14
Assemble Case	16
Software	30
Installing Arduino Libraries	30
Uploading Code	31
Host-Side Software	32
Using the UNTZtrument_Hello_World Example	33
Using the UNTZtrument_Step_Seq Example	33
Hacking	35
Hardware Considerations	35
Software Considerations	39
Using the UNTZtrument Arduino Library	40
Downloads	45

Overview

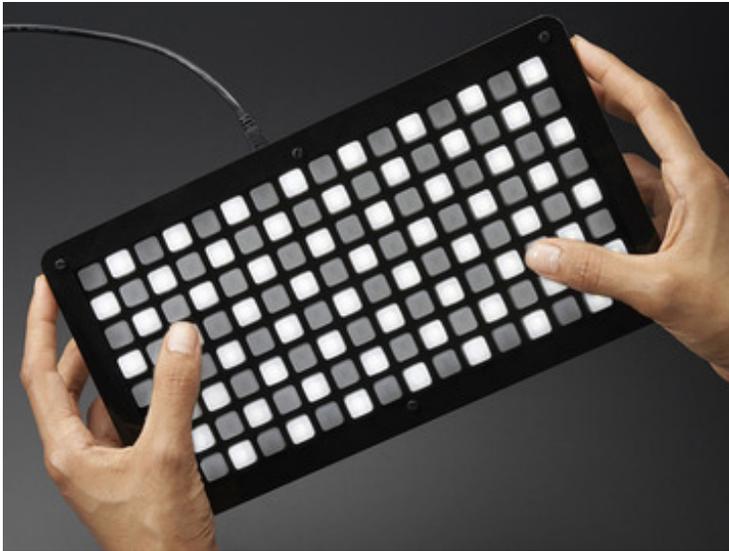


Build and customize your very own open-source button grid controller with **UNTZtrument!** This DIY kit comes with delicious translucent button pads, driver boards, diffused white LEDs and a custom laser cut enclosure. The result is a sturdy and elegant but also *super-hackable* controller for music, video...or something else???

We designed this kit for ease of use and ultimate flexibility. All you need is an Arduino Leonardo, basic soldering tools and an afternoon. Once assembled and programmed with the Arduino IDE, this box turns into a USB MIDI device that works with any computer and has 64 buttons (128 on the *HELLA UNTZtrument*) with individually-controllable LEDs. Our example programs send simple MIDI Note On and Off messages, but with a little programming ingenuity you can send and receive any kind of MIDI command. Since its USB MIDI it can work instantly with just about all synth software. Don't like MIDI? The Arduino

Leonardo can also emulate a USB keyboard or plain old USB serial.

Since it's Arduino-powered, adding more stuff like accelerometers, potentiometers, rotary encoders, etc. is straight-forward using existing libraries available on the Internets.

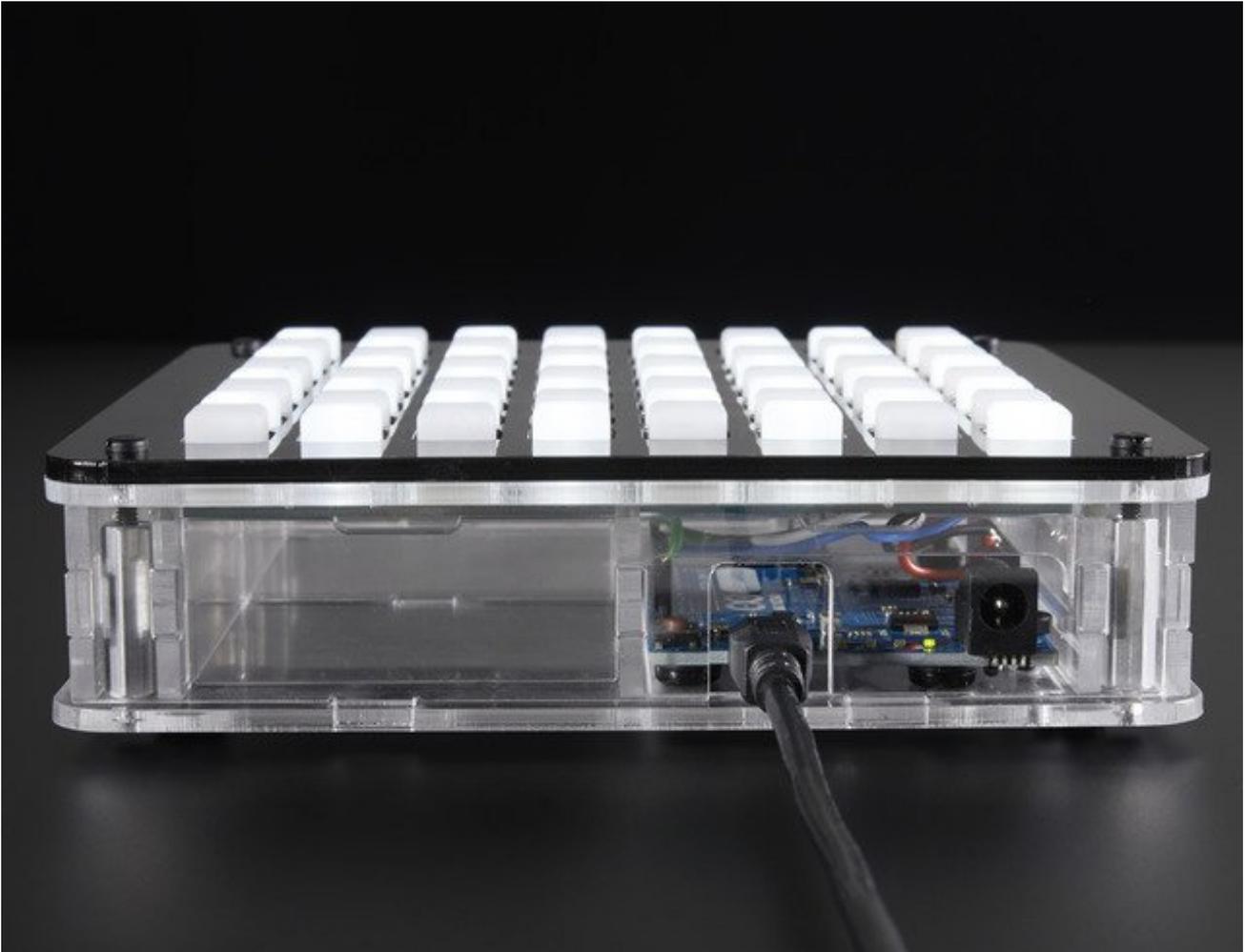


HELLA UNTZtrument ups the ante with **128** LED-backed buttons. It's *huge!*

What UNTZtrument *is:*



- UNTZtrument is a 64- or 128-button **MIDI device** that works alongside music software on your computer.
- It's a **kit**, requiring some **soldering**, a few tools and a little prior experience with the **Arduino** microcontroller.
- UNTZtrument is **open source**. The software is free and you can mix it up to add your own new features, or use code that others have written.



What UNTZtrument *is not*:

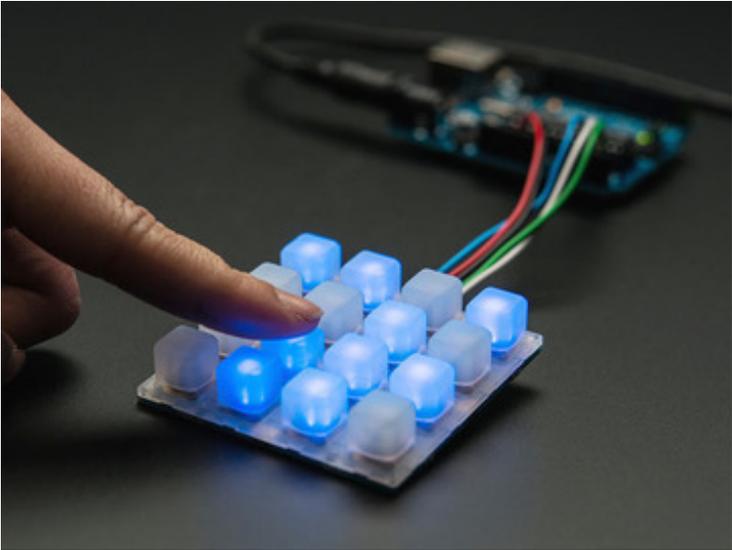
- UNTZtrument is *not* a self-contained musical instrument. It generates no audio and must be connected via **USB** to a computer to either create sounds or forward MIDI data to a synthesizer.
- UNTZtrument is *not* a Monome (or Arduinome), but looks similar. Those are *serial* USB devices requiring a software bridge to the Max visual programming language. UNTZtrument speaks **MIDI**, which is native to most music software.*

* UNTZtrument could certainly be *adapted* to be compatible with these...yay for open source...but we've found said bridge software to currently be quite finicky. Using MIDI now doesn't preclude using serial for other things later...we might revisit this as the situation evolves.



Here Collin Cunningham explains the basics of MIDI:

First Steps



UNTZtrument is based on Adafruit's *Trellis*, a 4x4 backlit keypad kit.

-



Four *Trellises* are combined to make a single large **8x8 matrix** for UNTZtrument. The HELLA UNTZtrument has *eight* *Trellises* in a 16x8 matrix!

Normally you have to buy three separate parts for *each* *Trellis* (PCB, elastomer keypad and LED pack), but the UNTZtrument kits have everything you need.

We selected **white** LEDs for the UNTZtrument kits. If you have a large stash of 3mm LEDs in some other color you can certainly use those instead.

-

You will also need an **Arduino Leonardo** microcontroller board, either the [regular](#)

[version](http://adafru.it/dy8) (http://adafru.it/dy8) or the [headerless variety](http://adafru.it/dy9) (http://adafru.it/dy9) if you want permanent connections.

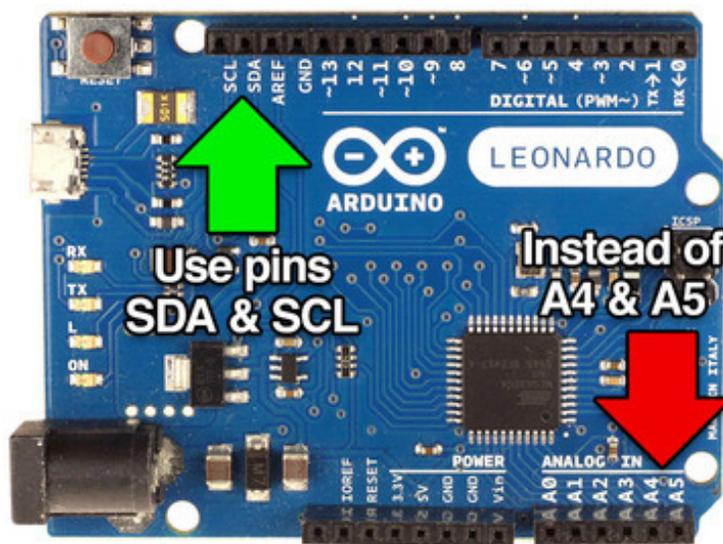
UNTZtrument will not work with the Arduino Uno, Mega or other boards. Must be the Arduino Leonardo, or a 100% compatible board based on the ATmega32U4 microcontroller. Because MIDI.

Arduino Leonardo. Period.

Additionally, you'll need some **wire** ([22 gauge solid-core](http://adafru.it/dya) (http://adafru.it/dya) wire is ideal, but stranded can work in a pinch), a **soldering iron & solder**, basic **hand tools** and a **Micro USB cable**.

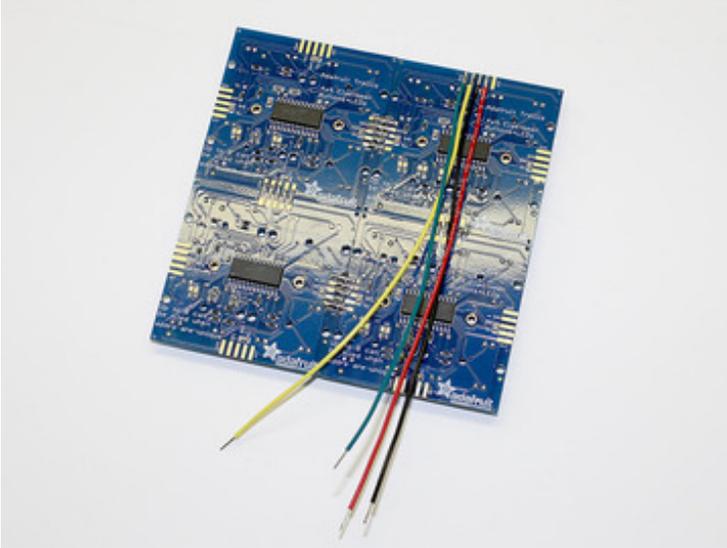
Let's Get Started!

So, as a first step to building your UNTZtrument kit, work through our [introductory Trellis guide](http://adafru.it/dxx) (http://adafru.it/dxx) first. **But with a few important changes:**



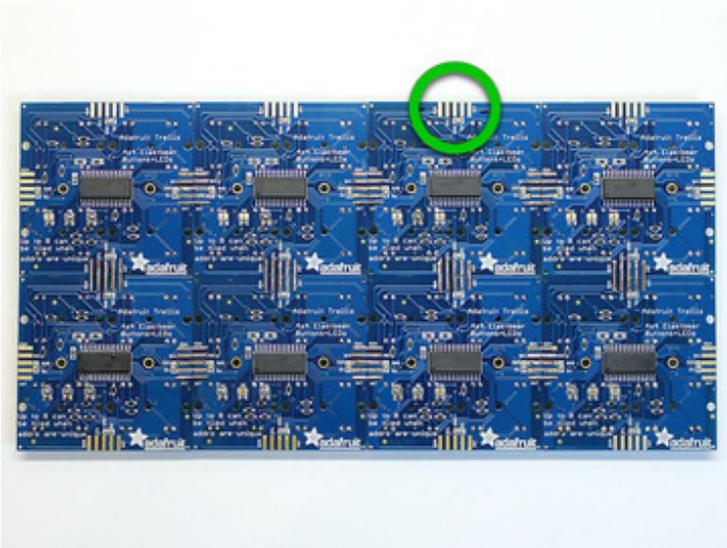
Because UNTZtrument is based around the **Arduino Leonardo**, it requires slightly different wiring: Use the **SDA** and **SCL** pins instead of A4 and A5 as shown in the guide.

Looking at the back of the tiled



Trellis boards...with the text upright, in the normal orientation for reading...connect the wires to the header along the **top edge**, toward the **right**. They should be about 6 inches (15 cm) long, or a little longer.

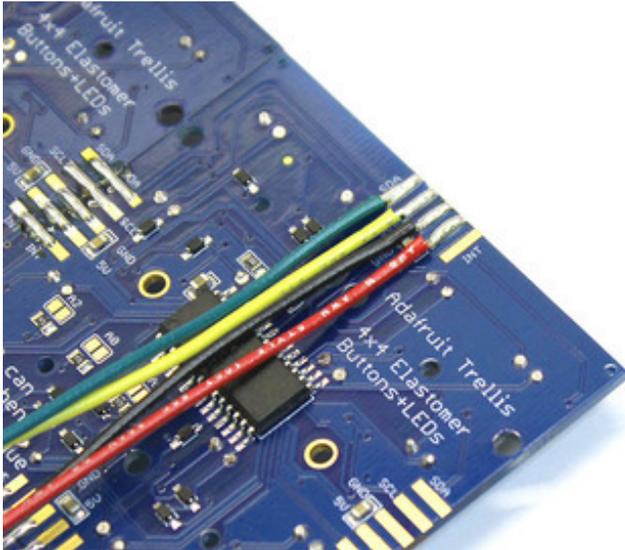
•



On the HELLA UNTZtrument — a 4x2 assembly of Trellises — connect the wires to the third header along the top edge, not the rightmost fourth header.

•

Try pointing the wires “inward” rather than off the edge of the board. This makes it easier to fit



in the case.

The INT pin is **not used** by UNTZtrument and does not need to be connected.



When setting the board addresses (by bridging the solder points on the back), use the values shown here.

Using a different order is *not* catastrophic, you'll just need to edit the code to match. Following this standard makes it easier to share code with other UNTZtrument users.

Check the orientation of the boards and use the large Adafruit silkscreen logo to make sure you have them oriented right and the correct addresses set.

Here's the address map for the HELLA UNTZtrument.



Use the code below for testing your UNTZtrument matrix rather than the code in the Trellis guide; it's already set up for the tiled matrix.

For the MELLA UNTZtrument, change NUMTRELLIS to 8.

```

/*****
This is a test example for the Adafruit Trellis w/HT16K33.
Reads buttons and sets/clears LEDs in a loop.
"momentary" mode lights only when a button is pressed.
"latching" mode toggles LED on/off when pressed.
4 or 8 matrices can be used. #define NUMTRELLIS to the
number in use.

```

Designed specifically to work with the Adafruit Trellis

----> <https://www.adafruit.com/products/1616>

----> <https://www.adafruit.com/products/1611>

Adafruit invests time and resources providing this open source code, please support Adafruit and open-source hardware by purchasing products from Adafruit!

Written by Limor Fried/Ladyada for Adafruit Industries.
MIT license, all text above must be included in any redistribution

*****/

```

#include <Wire.h>
#include "Adafruit_Trellis.h"

#define NUMTRELLIS 4 // **** SET # OF TRELLISES HERE

#define MOMENTARY 0
#define LATCHING 1

#define MODE LATCHING // **** SET MODE HERE

Adafruit_Trellis matrix[NUMTRELLIS] = {
  Adafruit_Trellis(), Adafruit_Trellis(),
  Adafruit_Trellis(), Adafruit_Trellis()
#if NUMTRELLIS > 4
  ,Adafruit_Trellis(), Adafruit_Trellis(),
  Adafruit_Trellis(), Adafruit_Trellis()
#endif
};

```

```

Adafruit_TrellisSet trellis = Adafruit_TrellisSet(
  &matrix[0], &matrix[1], &matrix[2], &matrix[3]
#ifdef NUMTRELLIS > 4
  ,&matrix[4], &matrix[5], &matrix[6], &matrix[7]
#endif
);

#define numKeys (NUMTRELLIS * 16)

// Connect Trellis Vin to 5V and Ground to ground.
// Connect I2C SDA pin to your Arduino SDA line.
// Connect I2C SCL pin to your Arduino SCL line.
// All Trellises share the SDA, SCL and INT pin!
// Even 8 tiles use only 3 wires max.

void setup() {
  Serial.begin(9600);
  Serial.println("Trellis Demo");

  // begin() with the addresses of each panel.
  // I find it easiest if the addresses are in order.
  trellis.begin(
    0x70, 0x71, 0x72, 0x73
#ifdef NUMTRELLIS > 4
    ,0x74, 0x75, 0x76, 0x77
#endif
  );

  // light up all the LEDs in order
  for (uint8_t i=0; i<numKeys; i++) {
    trellis.setLED(i);
    trellis.writeDisplay();
    delay(50);
  }
  // then turn them off
  for (uint8_t i=0; i<numKeys; i++) {
    trellis.clrLED(i);
    trellis.writeDisplay();
    delay(50);
  }
}

void loop() {
  delay(30); // 30ms delay is required, dont remove me!

  if (MODE == MOMENTARY) {
    // If a button was just pressed or released...
    if (trellis.readSwitches()) {
      // go through every button
      for (uint8_t i=0; i<numKeys; i++) {
        // if it was pressed, turn it on

```

```

if (trellis.justPressed(i)) {
  Serial.print("v"); Serial.println(i);
  trellis.setLED(i);
}
// if it was released, turn it off
if (trellis.justReleased(i)) {
  Serial.print("^"); Serial.println(i);
  trellis.clrLED(i);
}
}
// tell the trellis to set the LEDs we requested
trellis.writeDisplay();
}
}

if (MODE == LATCHING) {
  // If a button was just pressed or released...
  if (trellis.readSwitches()) {
    // go through every button
    for (uint8_t i=0; i<numKeys; i++) {
      // if it was pressed...
      if (trellis.justPressed(i)) {
        Serial.print("v"); Serial.println(i);
        // Alternate the LED
        if (trellis.isLED(i))
          trellis.clrLED(i);
        else
          trellis.setLED(i);
      }
    }
    // tell the trellis to set the LEDs we requested
    trellis.writeDisplay();
  }
}
}

```

So...with those changes in mind...here's a link to the starter guide:

[Introducing Adafruit Trellis \(http://adafru.it/dxx\)](http://adafru.it/dxx)

Don't continue with the UNTZtrument guide until you have a tested and working 8x8 or 16x8 Trellis.

Troubleshooting

Some of the LEDs don't light up!

- If the positions are somewhat random: some LEDs might have been installed backwards, or might've been damaged from excessive heat. Happens all the time, not to worry. This is why we include lots of spares. De-solder the problem LEDs and clean up the holes using a solder sucker, then replace them with new ones (in the correct orientation).
- If it's a complete quadrant of the 8x8 Trellis, the address jumpers on the back of the board might be improperly set. Refer to the diagram above.

None of the LEDs light up!

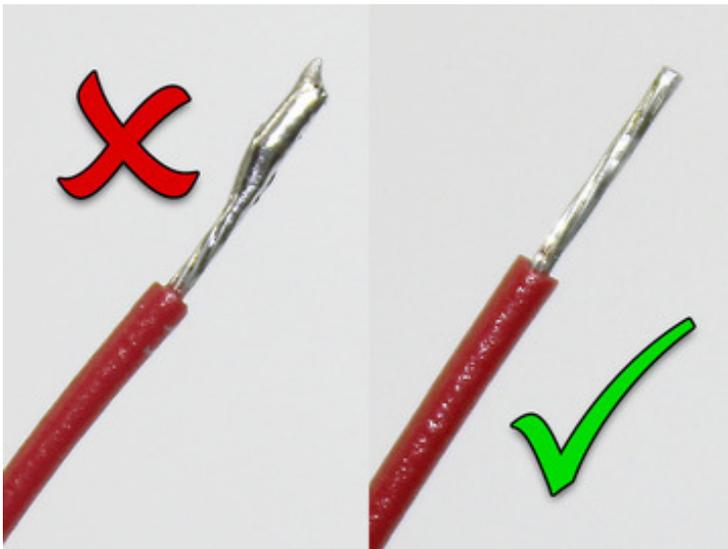
Might be the wiring. The Arduino Leonardo requires the use of different pins when communicating with the Trellis boards, or you might just have the wires swapped. Refer to the diagram above.

The wires keep snapping off the board!

Wires shearing off are usually due to *cold solder joints*, where the solder has not fully wetted the pad and flowed smoothly between the pad and wire. Make sure you're fully heating the pad and wire first *before* adding solder...*do not* melt solder on the tip of the iron and then carry it to the wire, that's a recipe for failure.

I can't fit the wires into the Arduino sockets!

It's a little easier to build UNTZtrument with solid-core wire (it slides right into the Arduino headers), but sometimes stranded is all you've got. Too-fat wires can happen if you've tinned the tip of a stranded wire with excessive solder...



When tinning stranded wires (to prevent fraying and shorts), use just a *tiny bit* of solder, and make sure it *fully flows* into the strands. Sometimes you'll want to clip off the very tip of the wire if the iron left a glop or spike there.

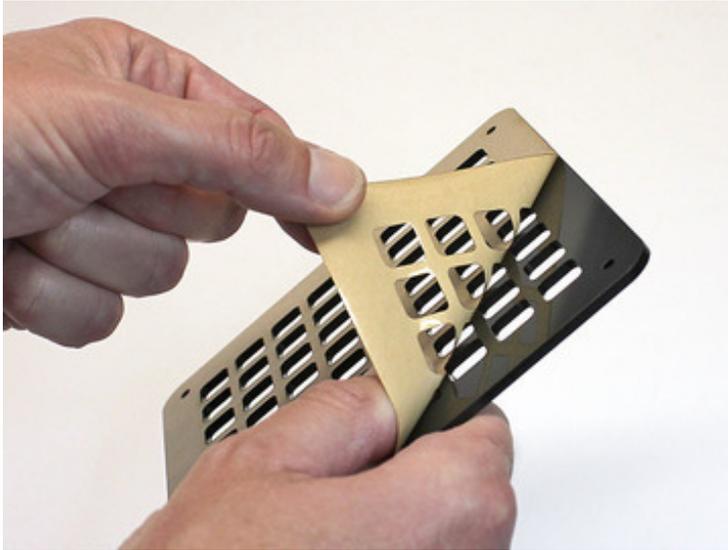
The [Adafruit Guide to Excellent Soldering](http://adafru.it/dxy) (<http://adafru.it/dxy>) has lots of advice for common soldering problems.

Reminder: don't proceed until you have a fully tested and working Trellis + Leonardo on

your desk. It's not easy to debug soldering once its inside the case!

Assemble Case

You should have a tested and fully working Trellis (without case) at this point. Don't proceed until you've reached that milestone.

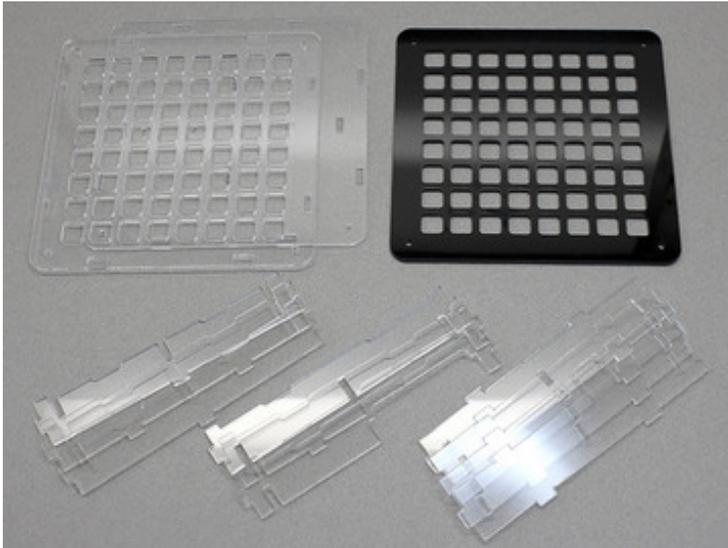


Start by peeling the backing paper off both sides of all the laser-cut parts. It's easiest to start at a corner, catching the edge of the paper with a fingernail.

The laser-cutting process sometimes leaves a little paper soot at the edges. If you like, you can wash these off with soap and water, just be **absolutely certain** that all the parts are **completely dry** before proceeding!

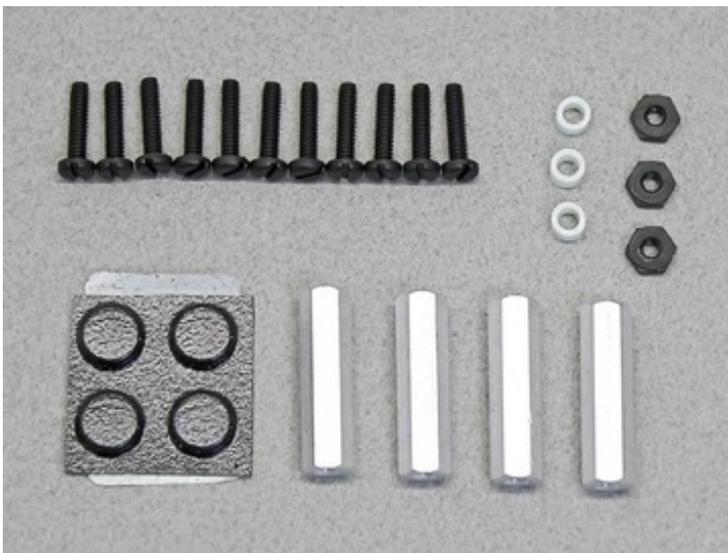
All told, there should be 13 laser-cut parts (15 for HELLA UNTZtrument). Most are clear except for the one black grid piece.

Your kit should also include the following hardware:



- Eleven (11) nylon screws
- Three (3) nylon nuts
- Three (3) 1/8" board spacers
- Four (4) 1" threaded standoffs
- One set of 4 peel-and-stick rubber feet

HELLA UNTZtrument has 2 more standoffs and 4 more screws for the wider case.



If any parts are missing or damaged, contact support@adafruit.com to arrange for a replacement.

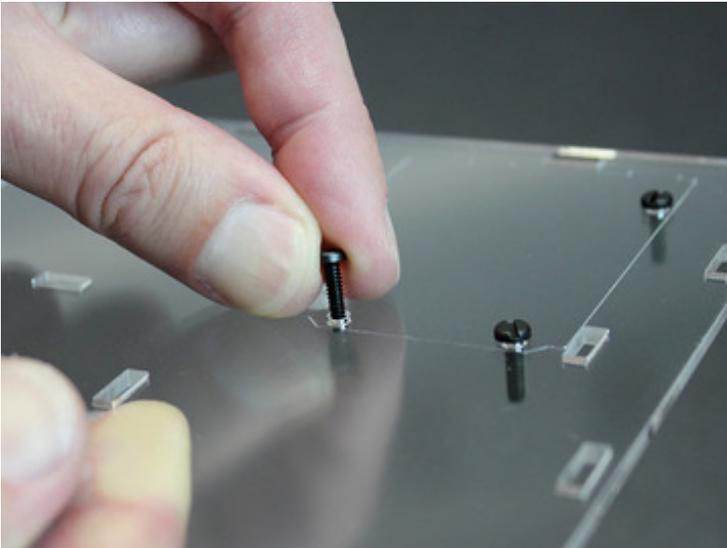
You've already assembled and tested the Trellis and Arduino Leonardo boards; they're implied but not listed in the above inventory.

We'll start with the base piece... it's one of the large squares, the one without the waffle grid.



This piece has the Arduino footprint scored on one face. This helps identify the top surface.

•



Turn the base over and install three screws in the Arduino footprint area.

This piece needs to be turned back over for additional work. You can either put a little masking tape over the head of each screw to hold them in place...or, if you're dextrous, grip the three screws from the other side as you turn it over.

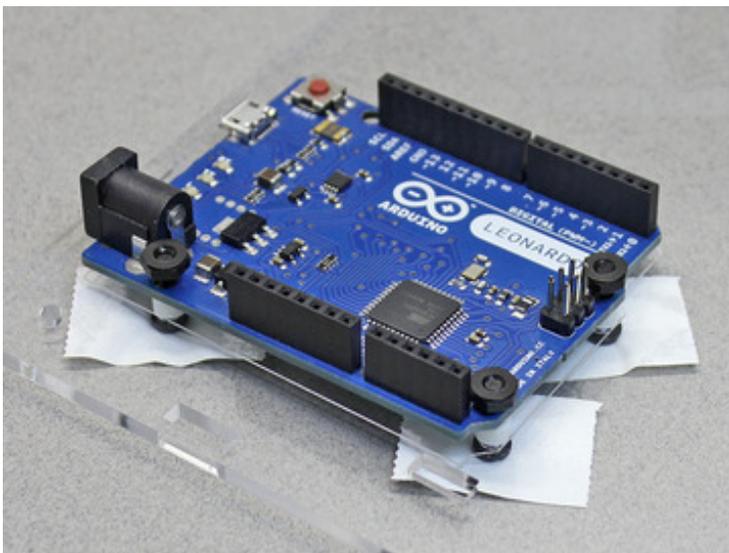
•



Set the base piece down so the screws are now pointed up.

Add a 1/8" nylon standoff over each screw.

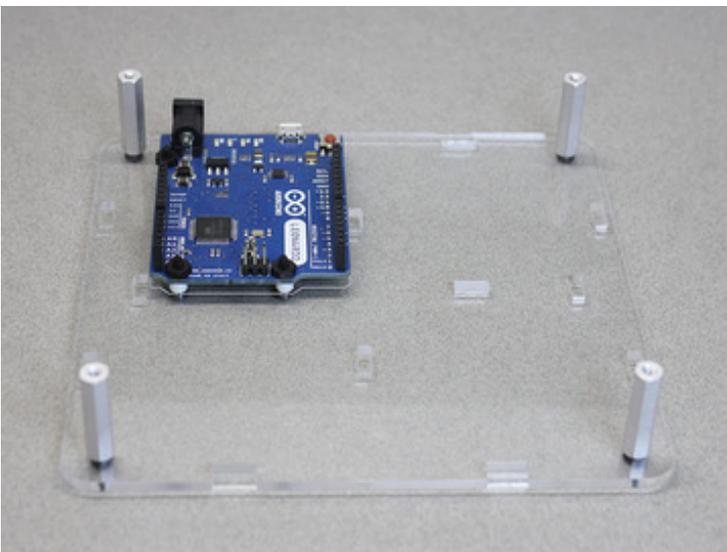
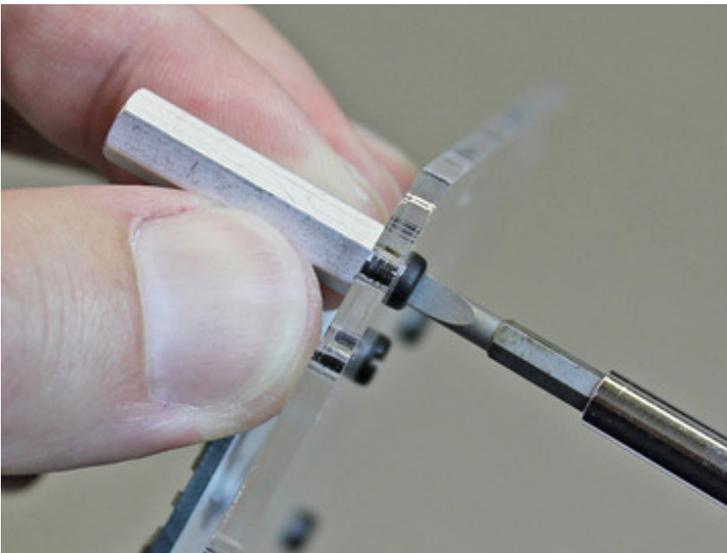
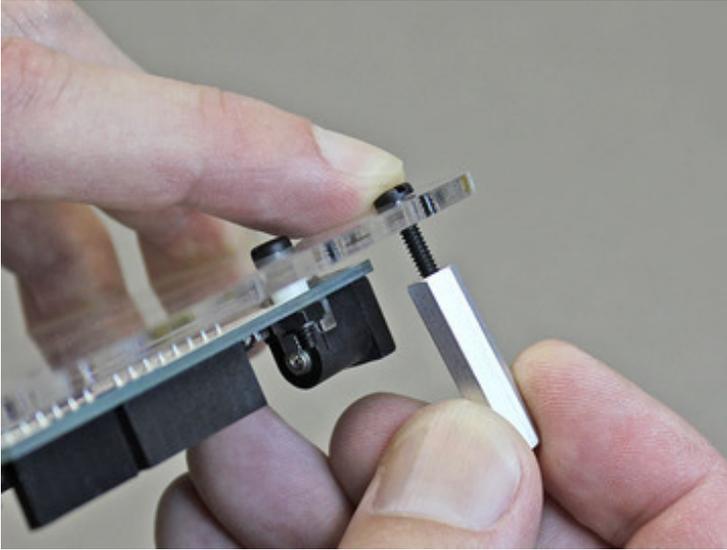
Install the Arduino board with the mounting holes over these three screws (there's a fourth hole, but we're not using it here).



Add a nut to each screw. There will probably be some mechanical interference from nearby headers and parts...that's okay, you just need to get the nuts started.

Now turn the base over, remove the tape (if used) and gently tighten the three screws with a small screwdriver.





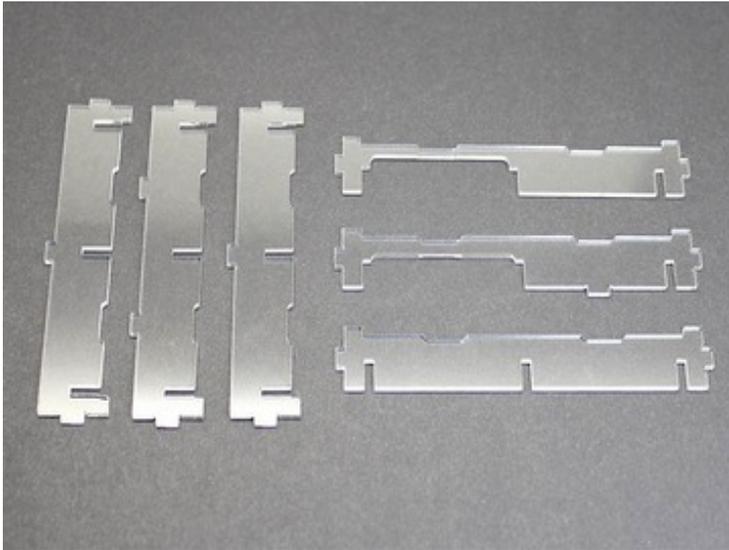
The standoffs are easier, we can do them one at a time. No need for tape.

Insert a screw into one of the four corner holes. Come up from the underside, as you did with the Arduino. Catch the screw in the threads of the standoff and turn it into place. Finger pressure is usually sufficient, or you can gently use a screwdriver.

Repeat until all four standoffs are installed (six for HELLA UNTZtrument). You should then be able to set the base down with

the Arduino and standoffs all on the top side.

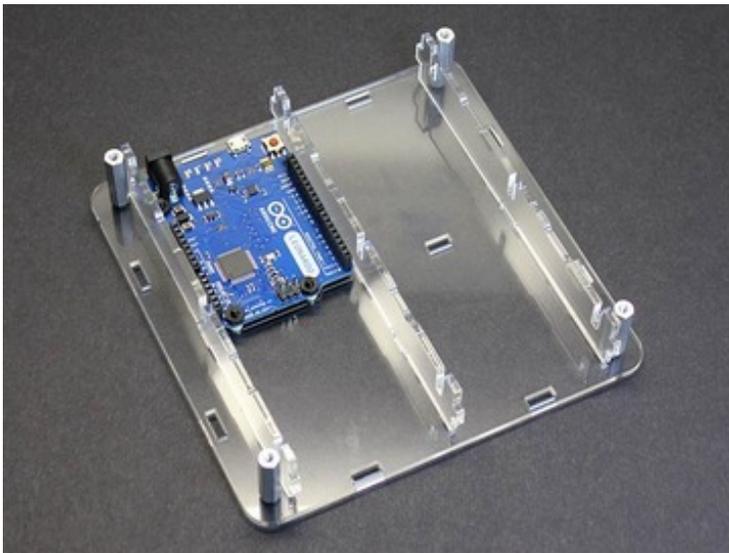
Next we'll install the vertical and



horizontal braces.

The vertical braces (which run parallel to the Arduino's longer axis) all look similar, but notice the "nubs" on the bottom are different: one, two, one. These fit into corresponding slots on the base. (On the HELLA UNTZtrument, there are five vertical braces — again, they're keyed to only fit specific positions).

•

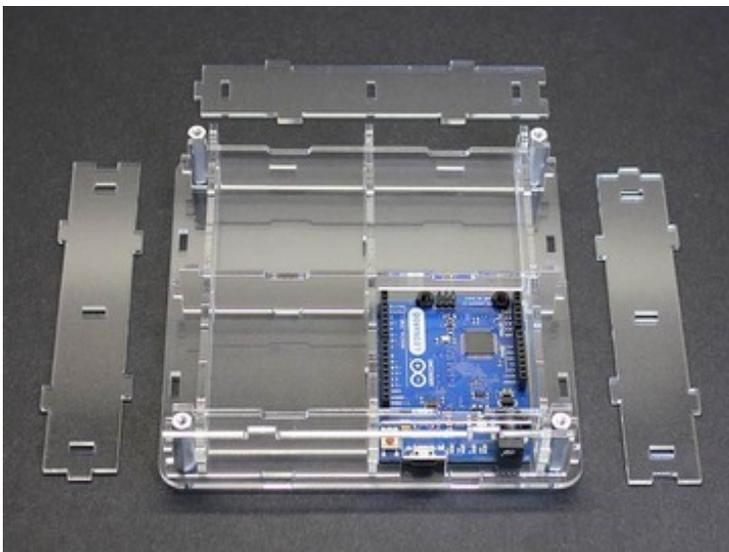
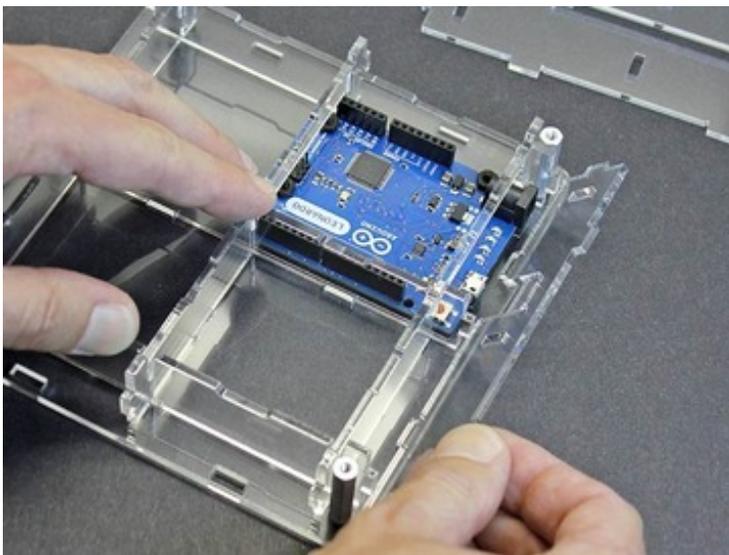
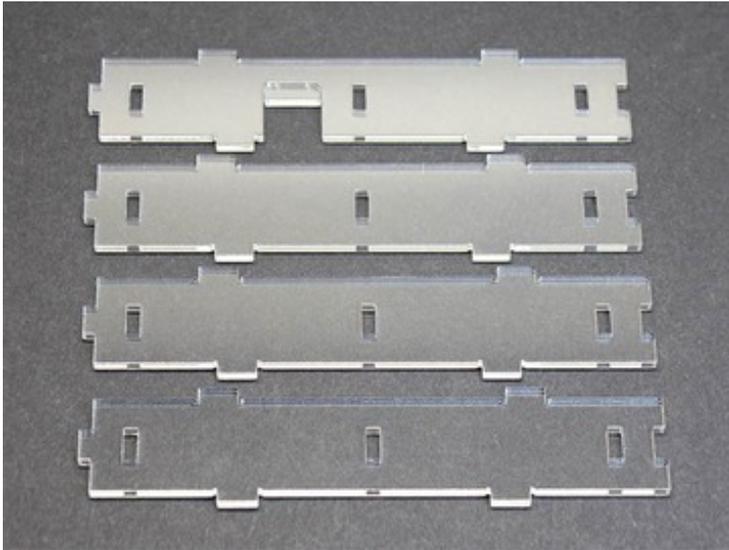


The horizontal braces now slot into notches on the verticals. Each of these is a different shape. The thinnest one is designed to clear the Arduino's power jack. The slightly thicker (but still pretty thin) brace goes down the middle, while the full-thickness one is at the bottom.

•

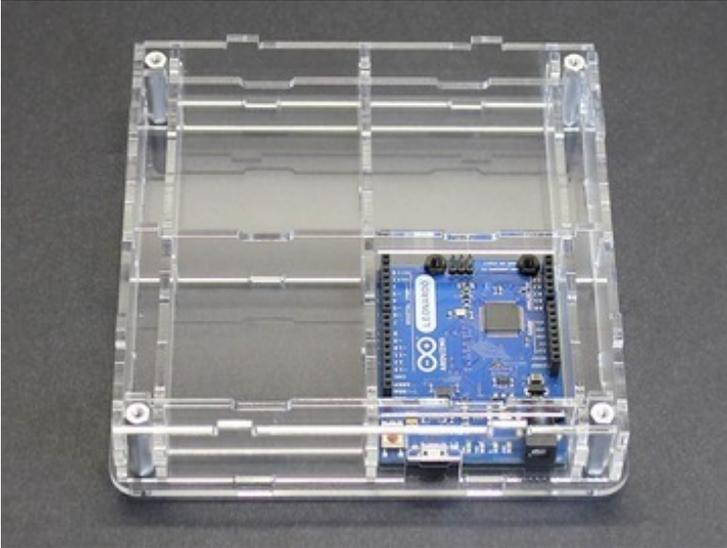


•



The four side pieces are *almost* identical...except for one which has a notch for the Arduino's USB port. Install this side first. Align the bottom tabs with the slots in the base and tilt it up into place. If the USB port is covered, you've got it backwards — turn it around and try again.

Once the first side is in place, the alignment of the remaining three should be straightforward.



Now get the Trellis board ready. Remove the rubber buttons for the time being and turn the board face-down.

Orientation is important. Once installed, the top edge of the Trellis will face be aligned with the Arduino's USB port.

Insert the power and signal wires from the Trellis to the headers on the Arduino:

- 5V from Trellis to 5V on Arduino
- GND from Trellis to any GND on Arduino
- SCL from Trellis to SCL on Arduino (this pin is nearest the mounting hole with no screw through it)