

Learn to Solder with ArduTouch Music Synthesizer kit and make music, sound, and noise!

Mitch Altman

Chief Scientist, **Cornfield Electronics**, San Francisco, CA

Inventor of **TV-B-Gone** universal remote controls

Co-founder of **3Ware** (successful Silicon Valley startup)

Pioneer of **VR** (in the mid-1980s)

Founding mentor at **HAX** (1st and biggest hardware accelerator)

Co-founder of **Noisebridge** (San Francisco hackerspace)

email: mitch@CornfieldElectronics.com

site: www.CornfieldElectronics.com

facebook: [maltman23](https://www.facebook.com/maltman23)

flickr: [maltman23](https://www.flickr.com/photos/maltman23)

WeChat: [mitchaltman](https://wechat.com/mitchaltman)

Fediverse: [@maltman23@mastodon.social](https://maltman23@mastodon.social)

Patreon: [mitchaltman](https://www.patreon.com/mitchaltman)

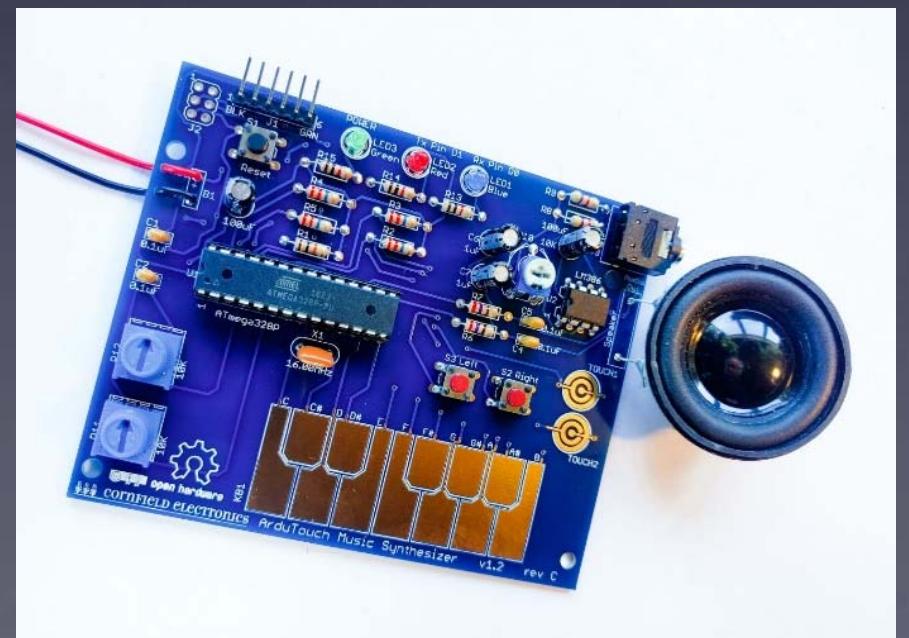


cornFIELD electronics

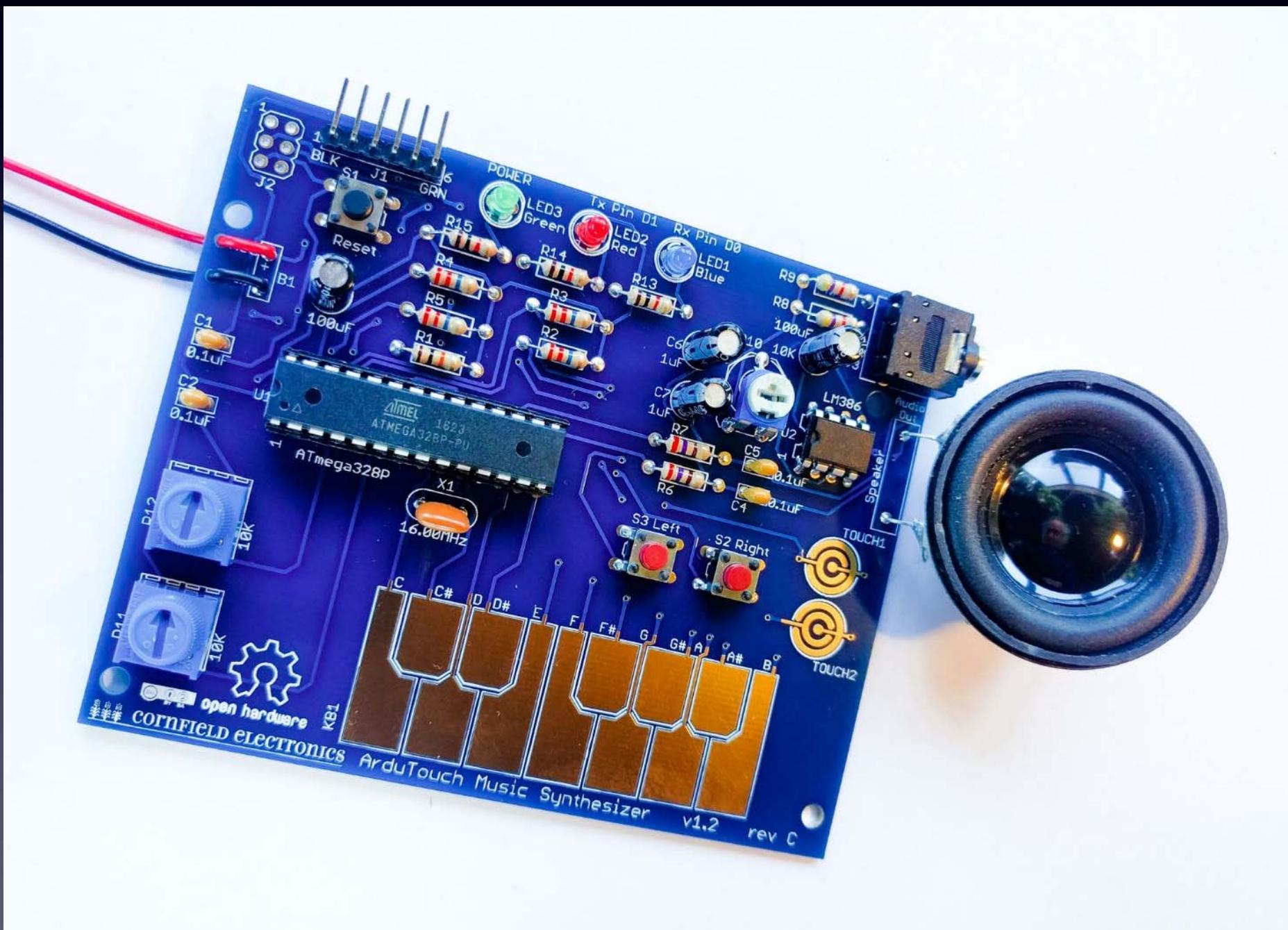
Syllabus

- Intro to ArduTouch music synthesizer kit
- Live demo of ArduTouch
- Intro to music synthesis / Digital Signal Processing
- How to solder
- How to program ArduTouch
with Arduino software

Soldering Workshops / kits

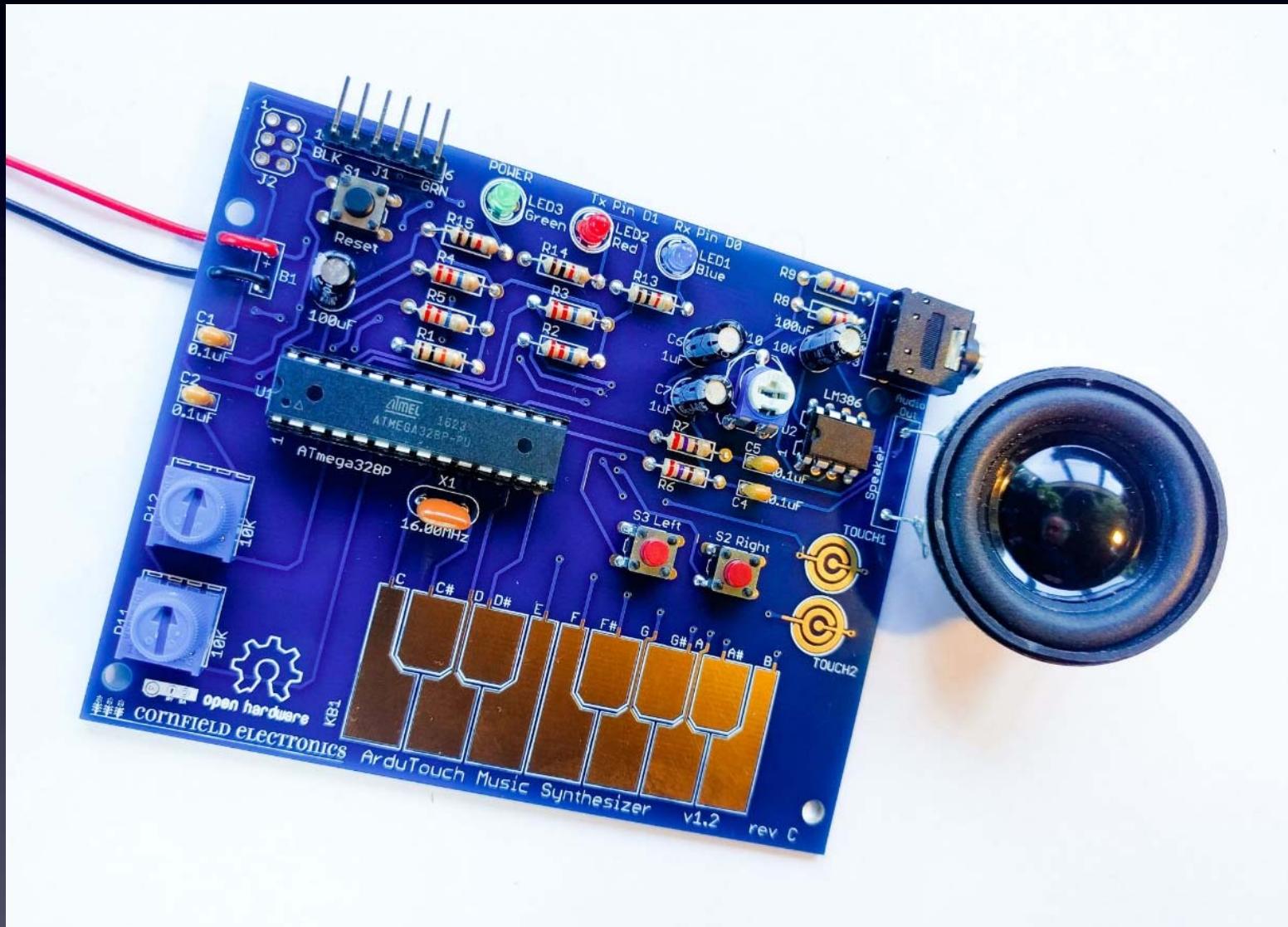


ArduTouch Music Synthesizer

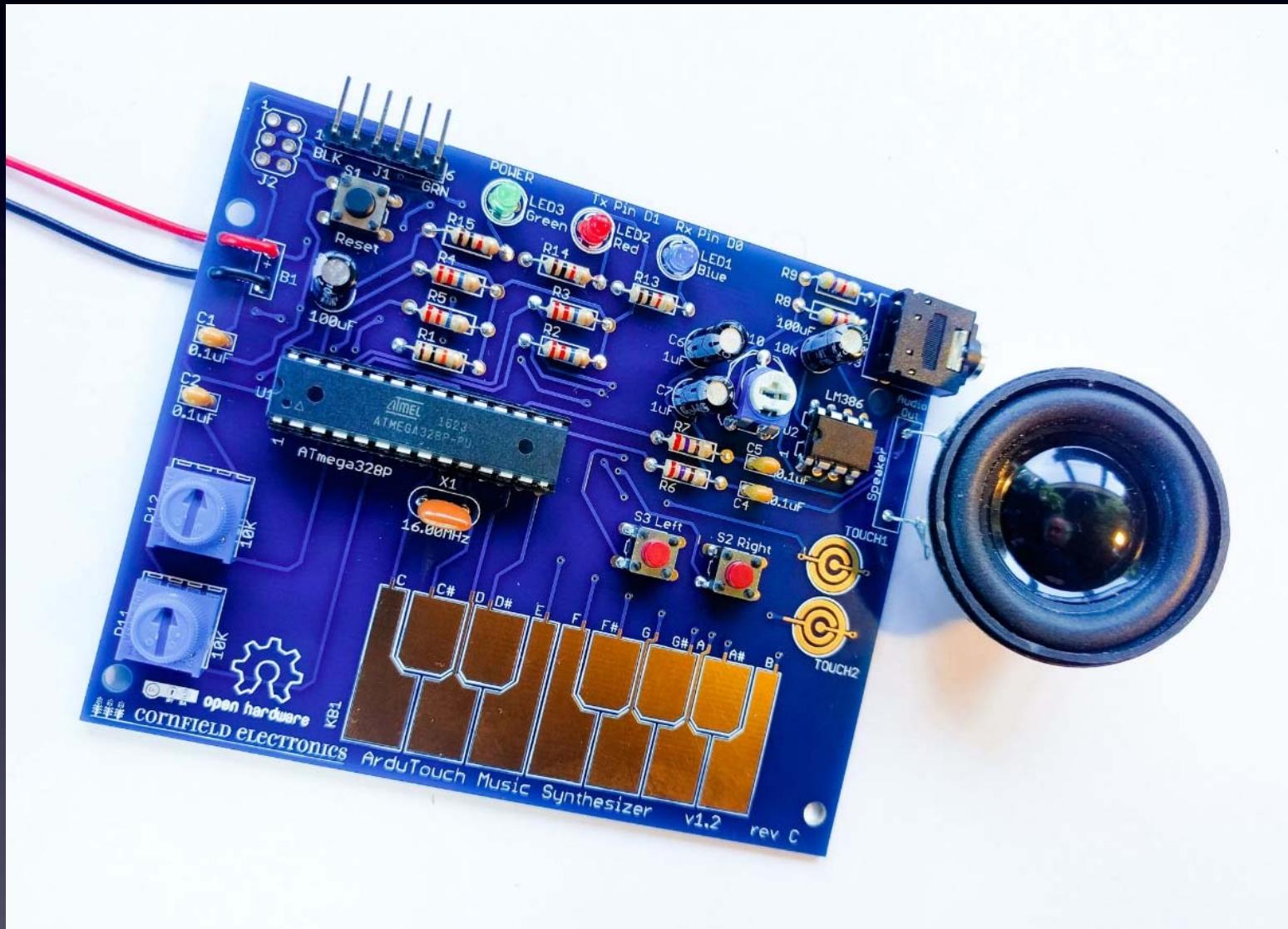


rev C

ArduTouch

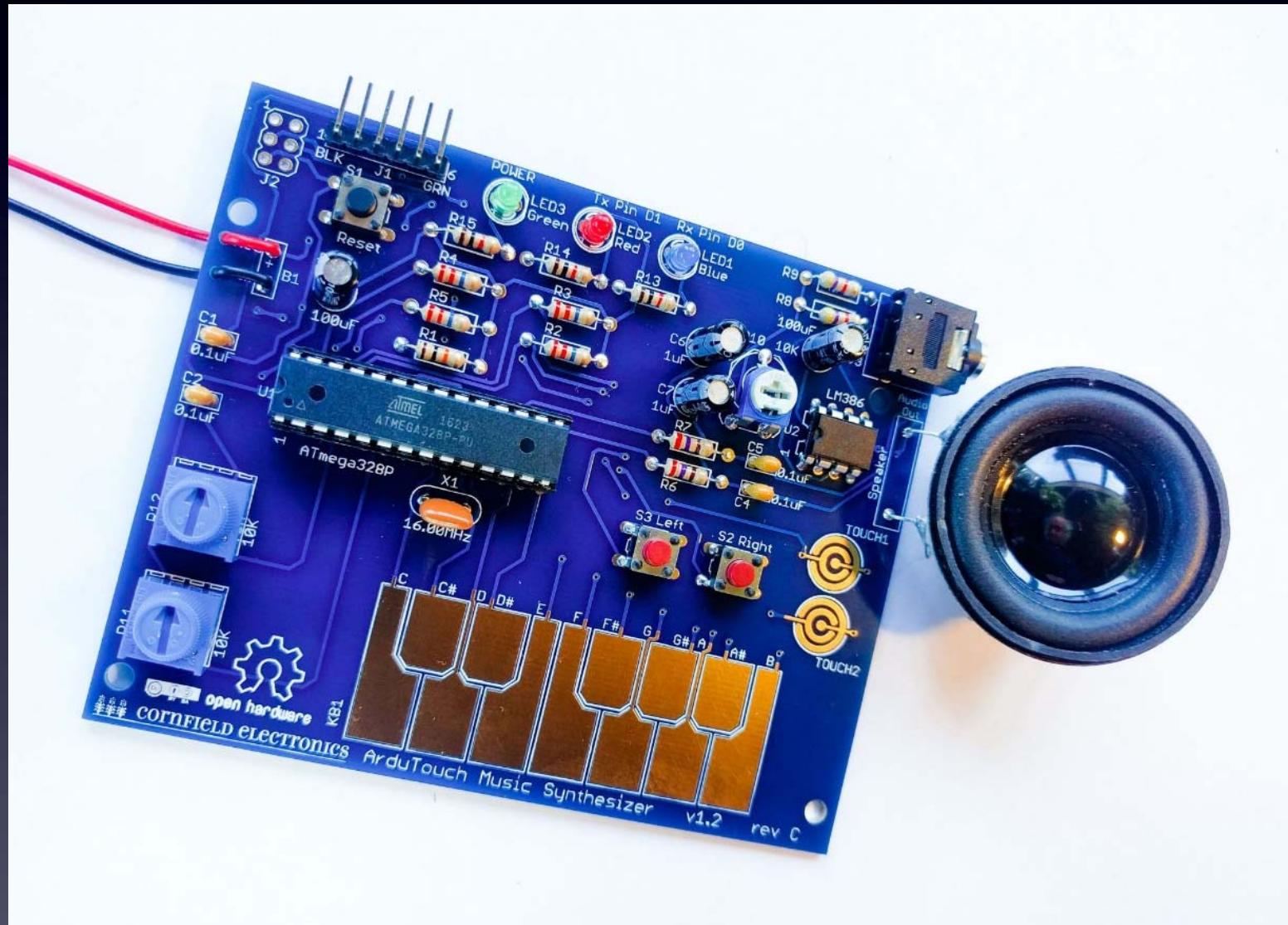


ArduTouch



Great for
learning
to solder

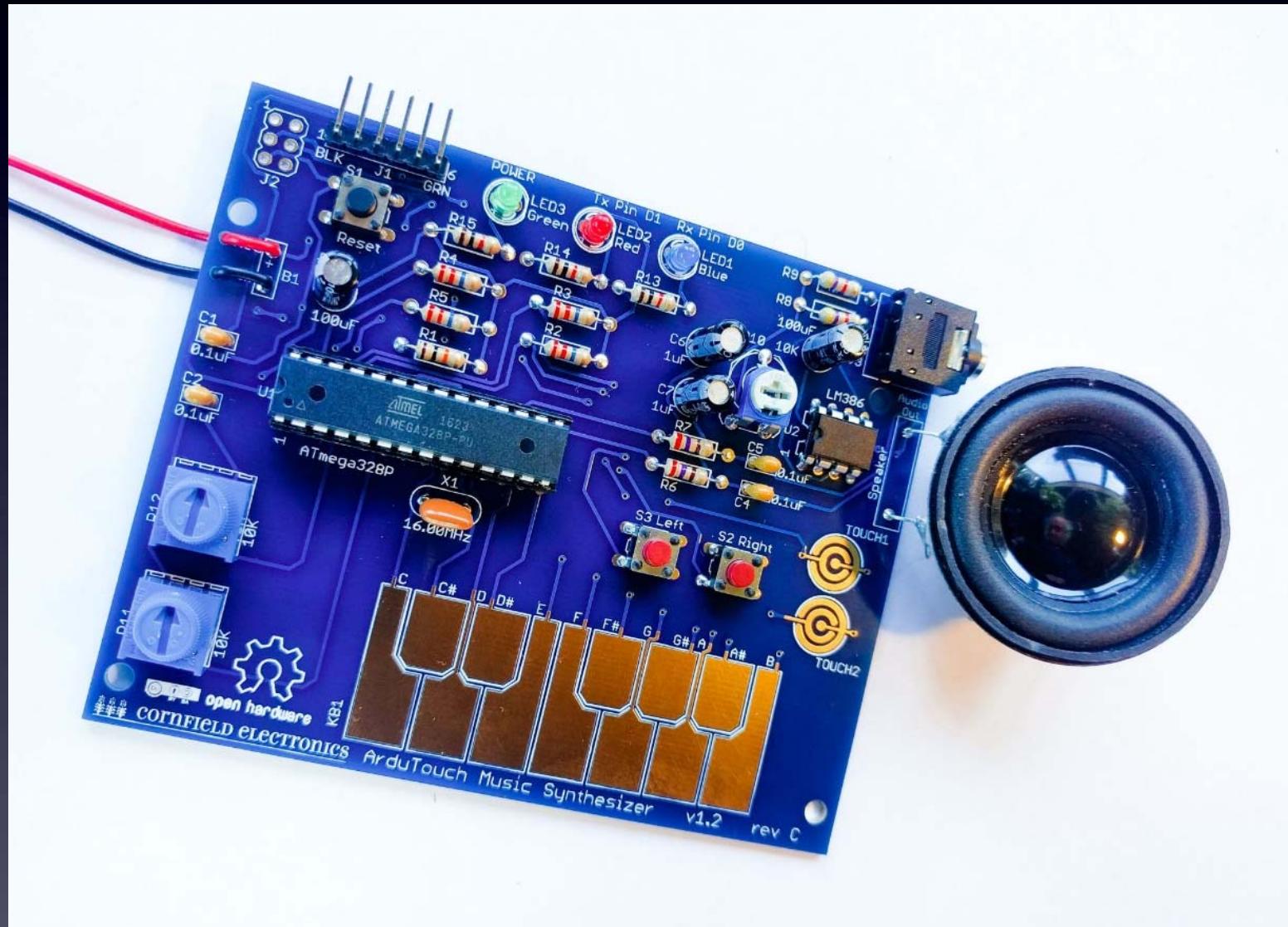
ArduTouch



**Solder it together
– and it works!**

*And you can also
program
your own synthesizers*

ArduTouch

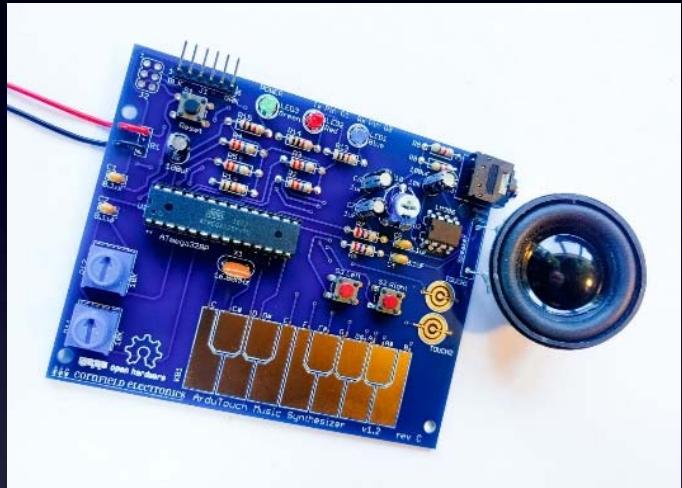


Solder it together
– and it works!

And you can also
program
your own synthesizers

You can also
learn
Digital Signal Processing

ArduTouch



Live demo

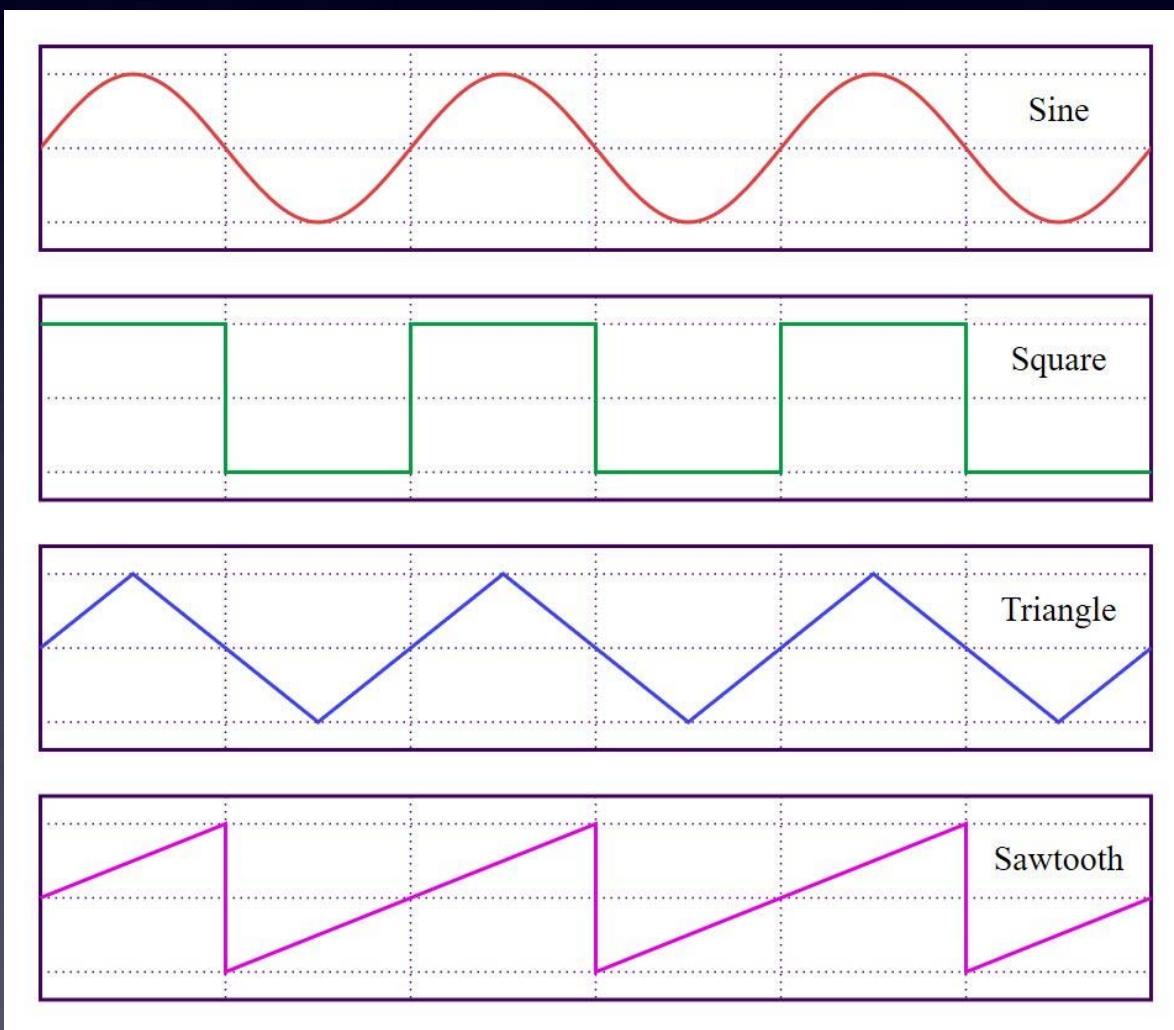
Some Types of Synthesizers

Analog



Some Types of Synthesizers

Analog



Modular Analog Synthesizer:

- Basic waveform oscillators
- Filters (to muck with sound)

Some Types of Synthesizers

Digital



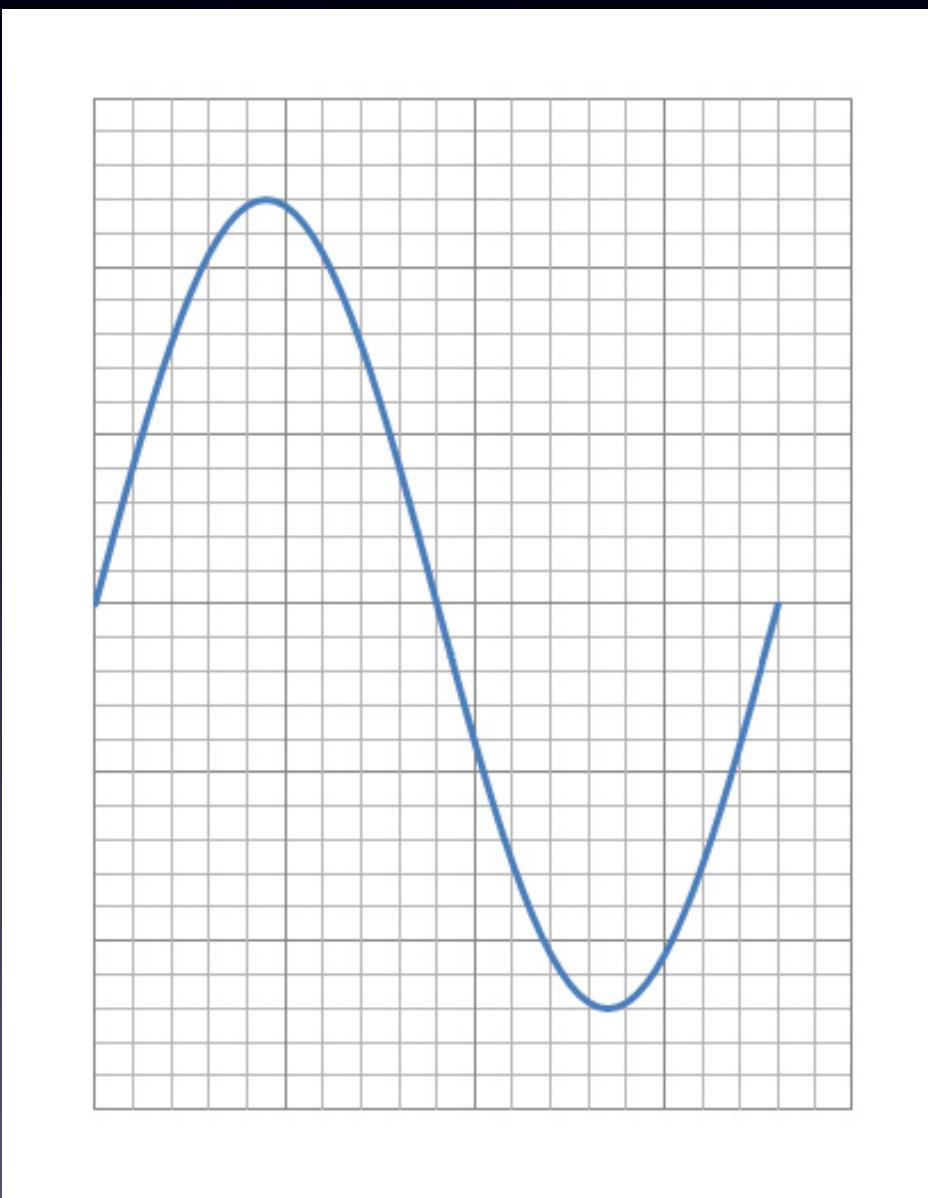
Some Types of Synthesizers

Digital

- Break things into little bits (or create little bits)
- Mess with it
- Put it back together again

Digital Signal Processing

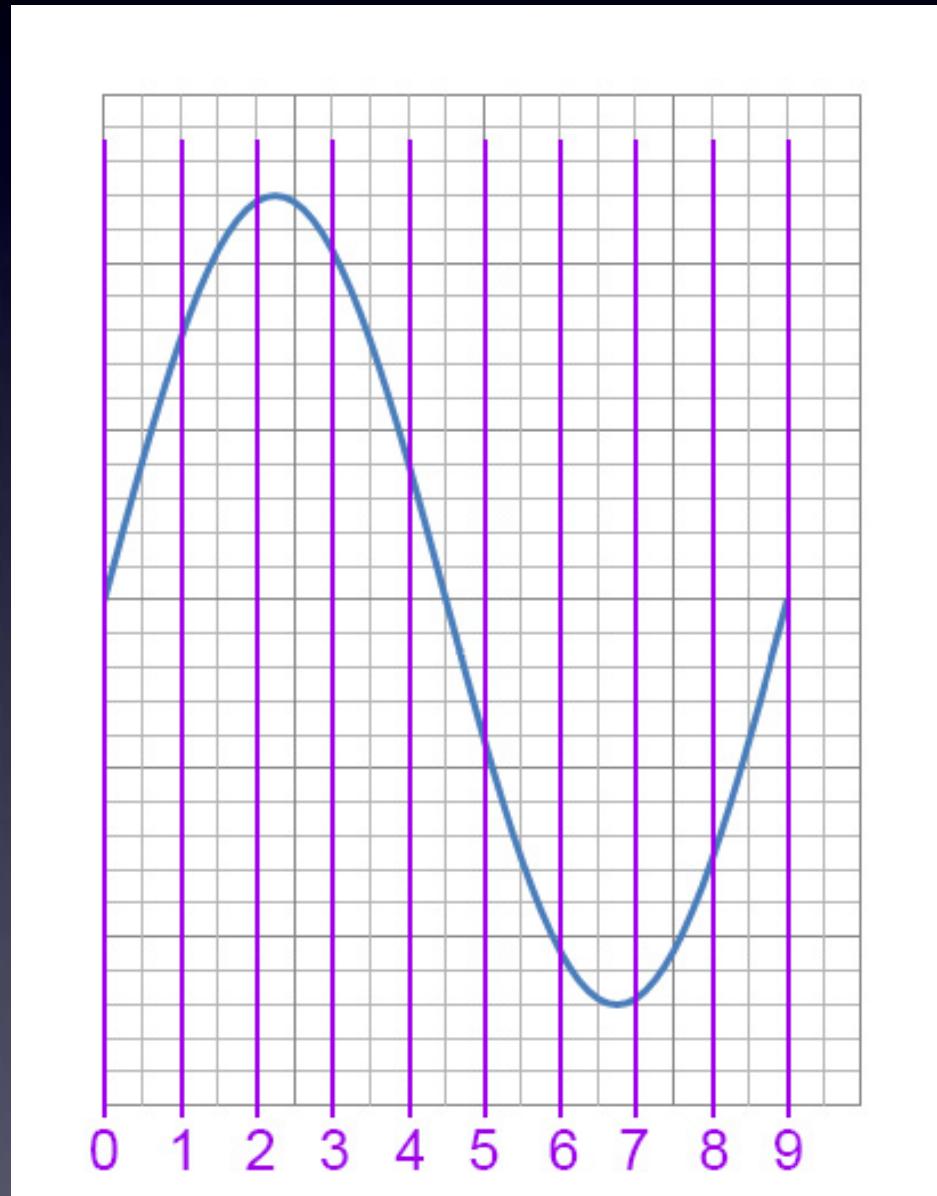
Analog waveform



Digital Signal Processing

To record it digitally

First slice it
(equal time slices)

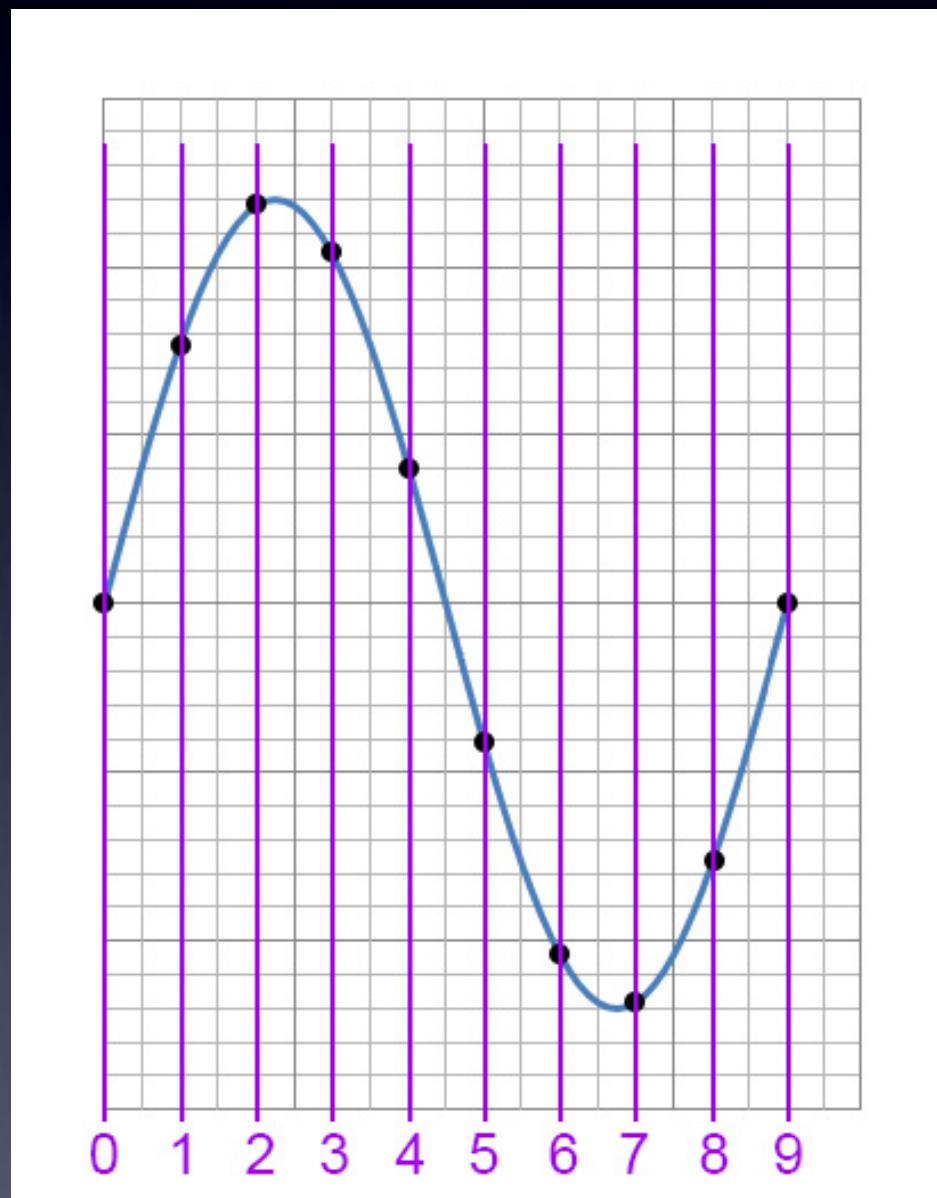


Digital Signal Processing

To record it digitally

First slice it

Then get the values

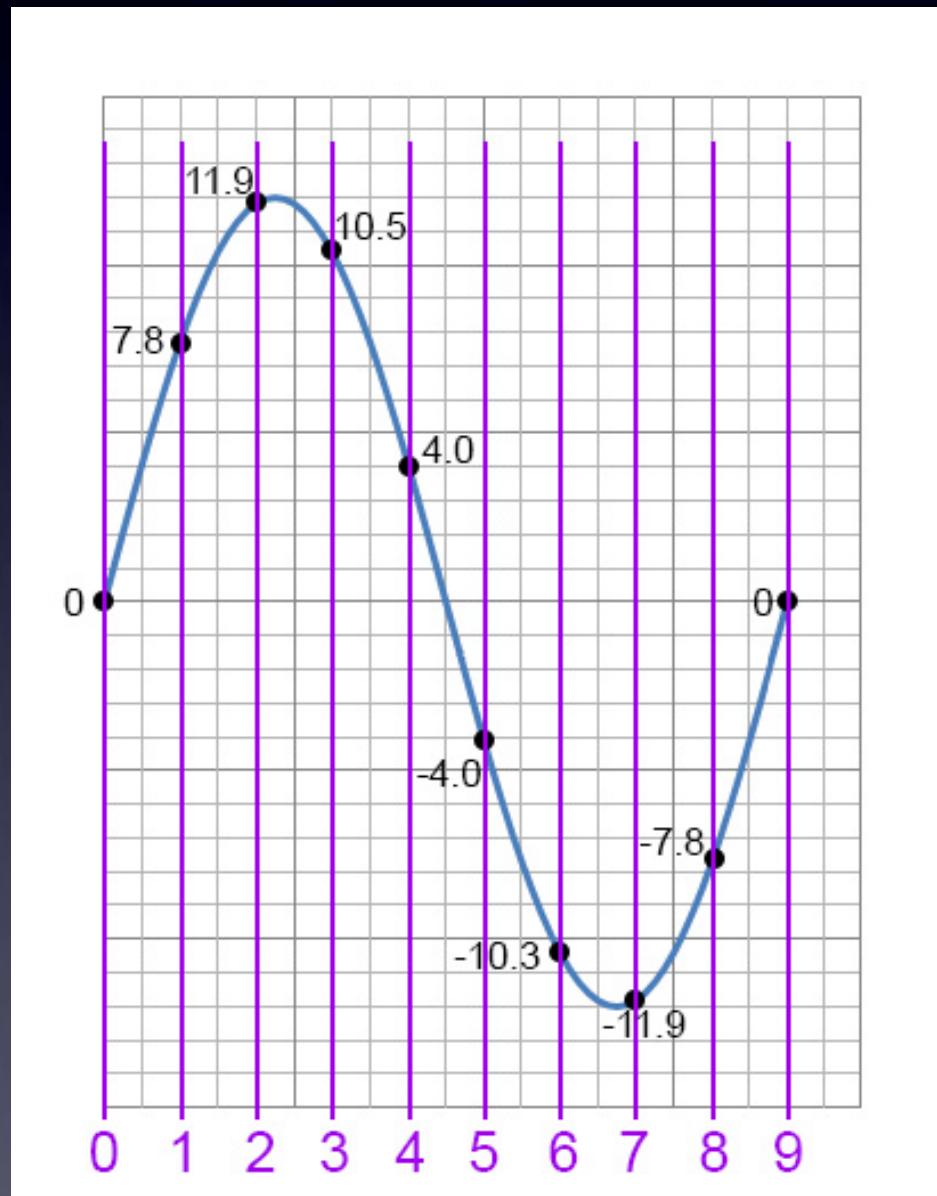


Digital Signal Processing

To record it digitally

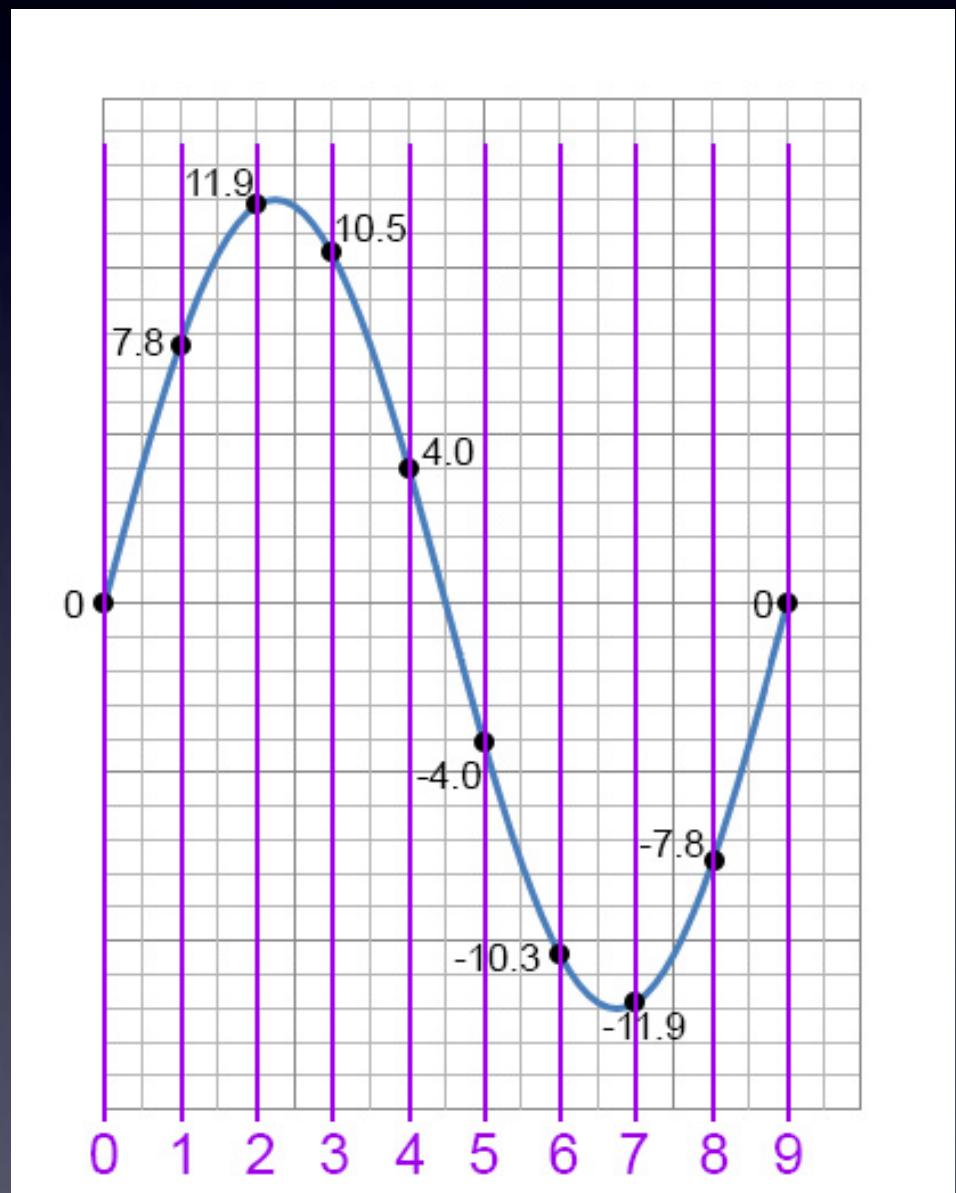
First slice it

Then get the values



Digital Signal Processing

To record it digitally
First slice it
Then get the values
Then store the values



Sample #	Digitized Value
0	0.0
1	7.8
2	11.9
3	10.5
4	4.0
5	-4.0
6	-10.3
7	-11.9
8	-7.8
9	0.0

Digital Signal Processing

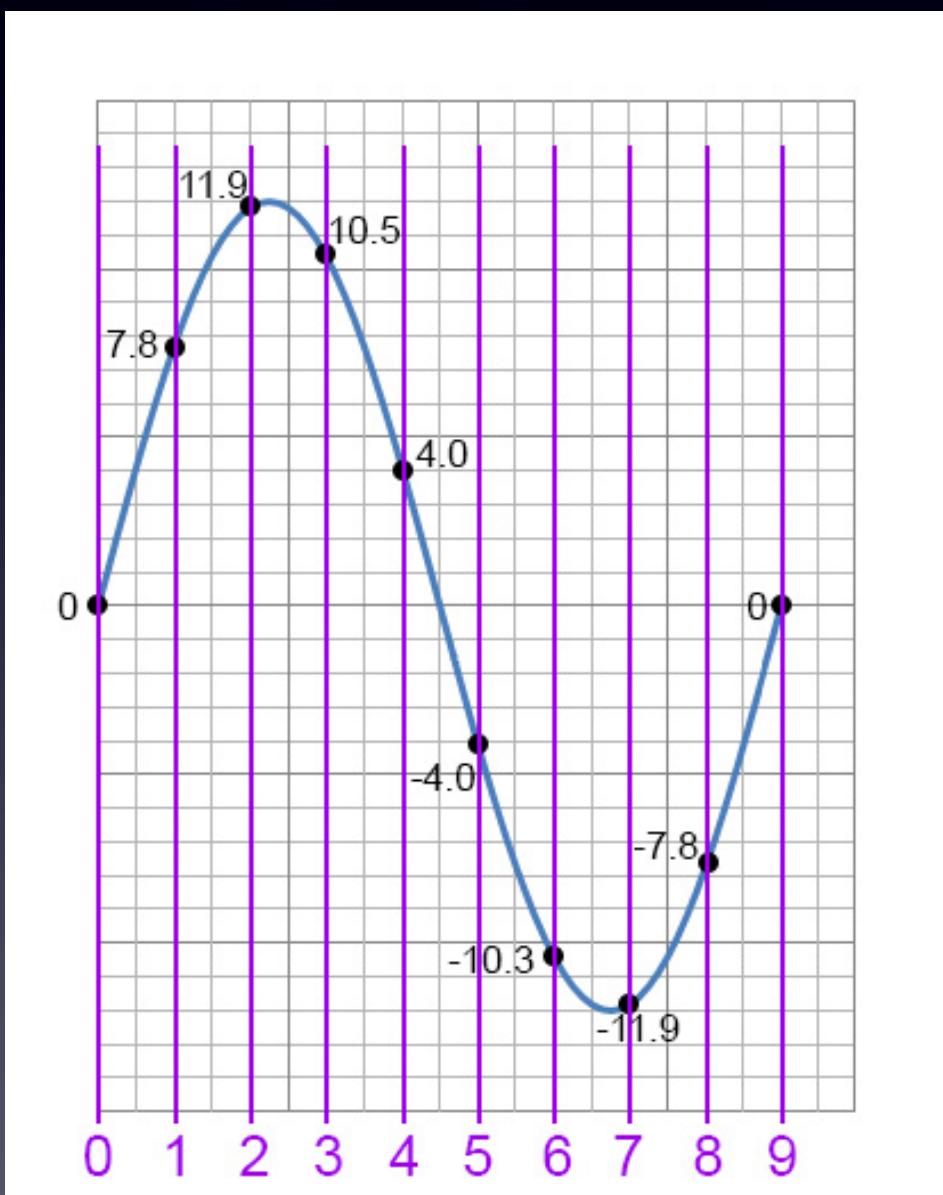
To record it digitally

First slice it

Then get the values

Then store the values

This is called: “Sampling”



Digital Signal Processing

Samples stored
in
Memory

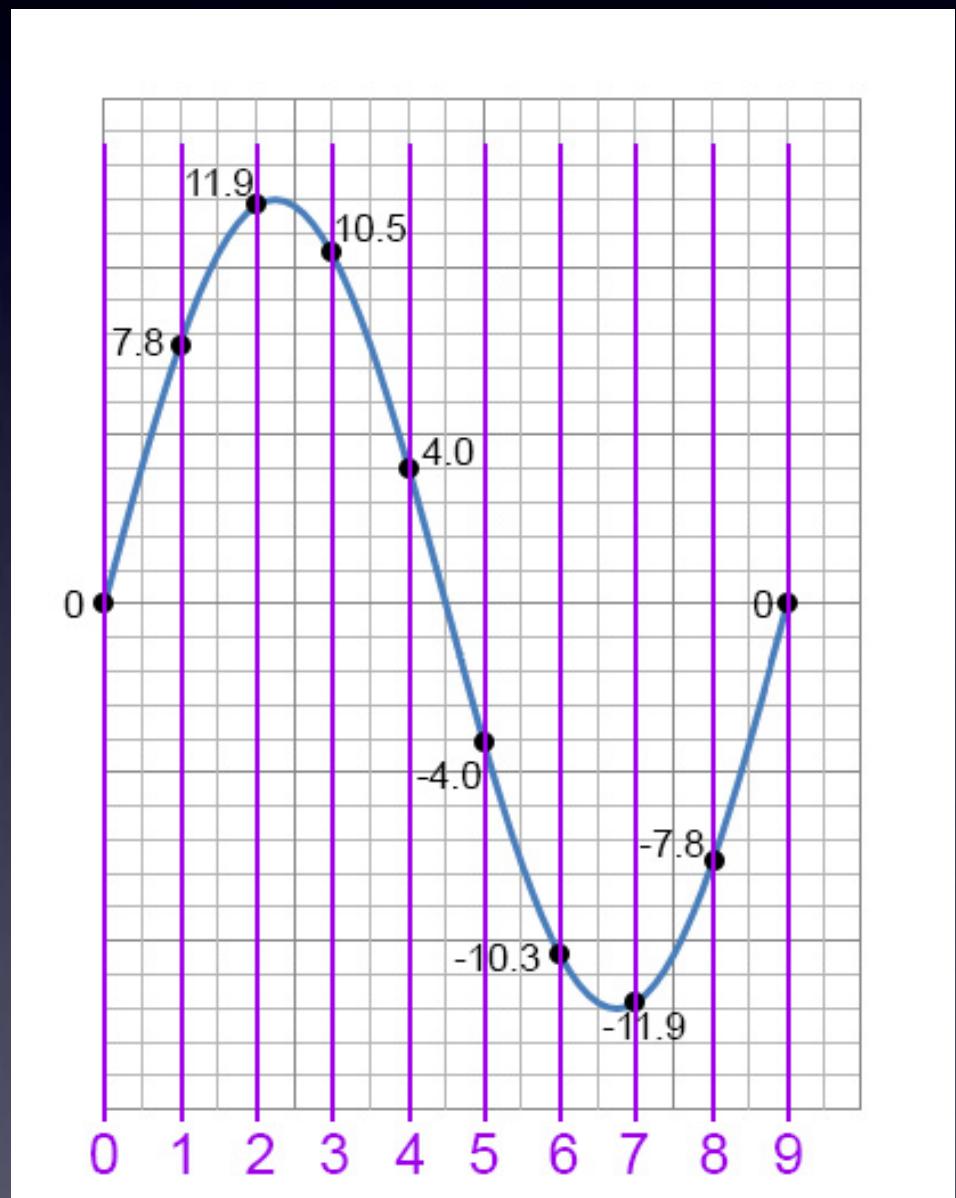
To record it digitally

First slice it

Then get the values

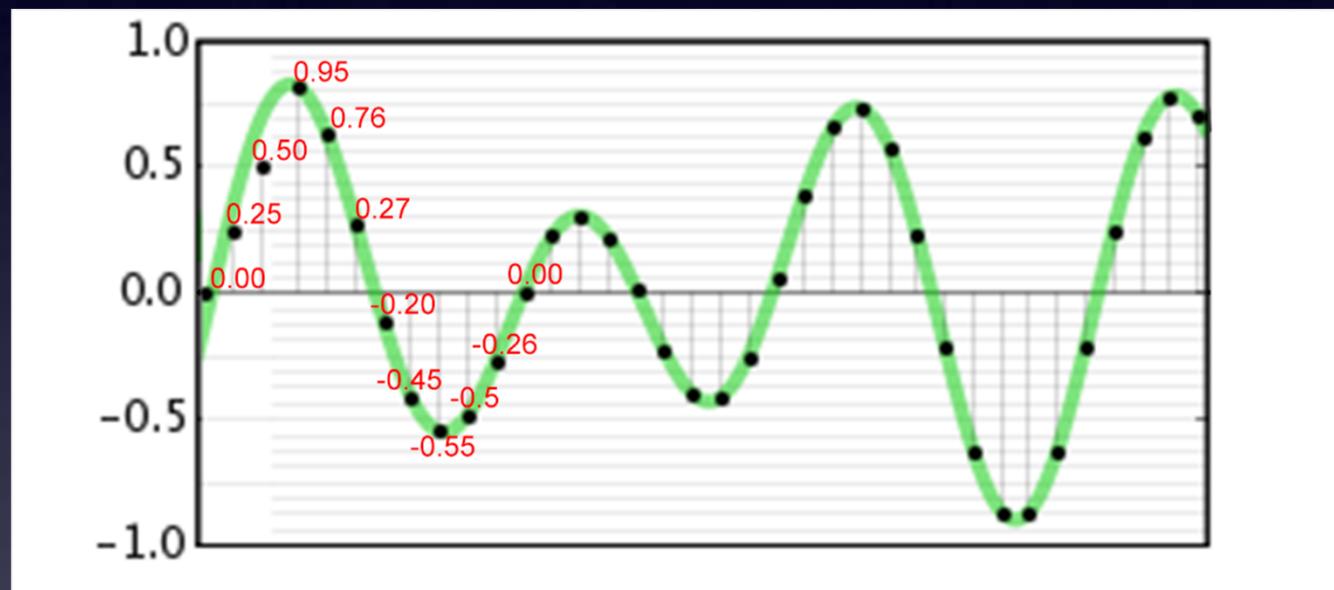
Then store the values

Waveform is:
“Digitized”



Memory loc	Memory contents
0	0.0
1	7.8
2	11.9
3	10.5
4	4.0
5	-4.0
6	-10.3
7	-11.9
8	-7.8
9	0.0

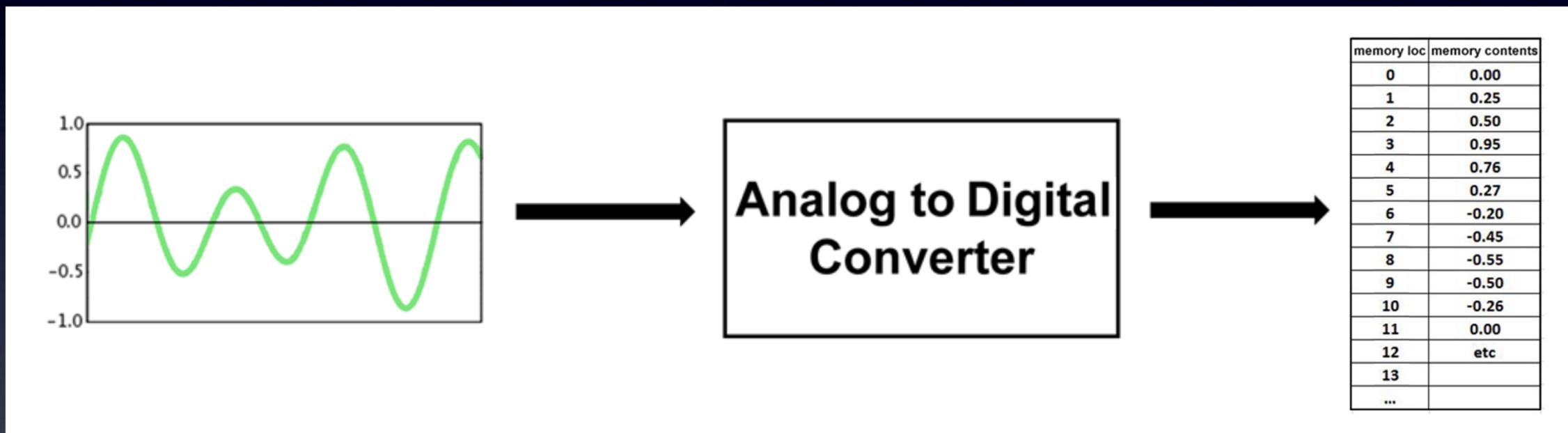
Digital Signal Processing



memory loc	memory contents
0	0.00
1	0.25
2	0.50
3	0.95
4	0.76
5	0.27
6	-0.20
7	-0.45
8	-0.55
9	-0.50
10	-0.26
11	0.00
12	etc
13	
...	

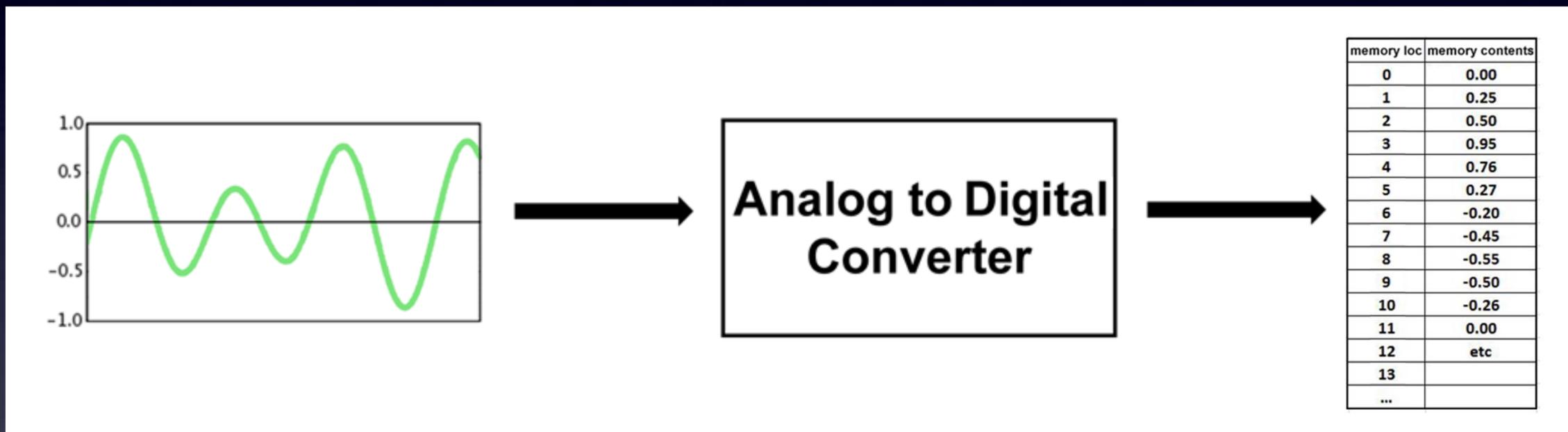
Digitized waveform can be any soundwave

Digital Signal Processing



Analog to Digital Conversion:
sampling an analog waveform to store it in digital memory

Digital Signal Processing



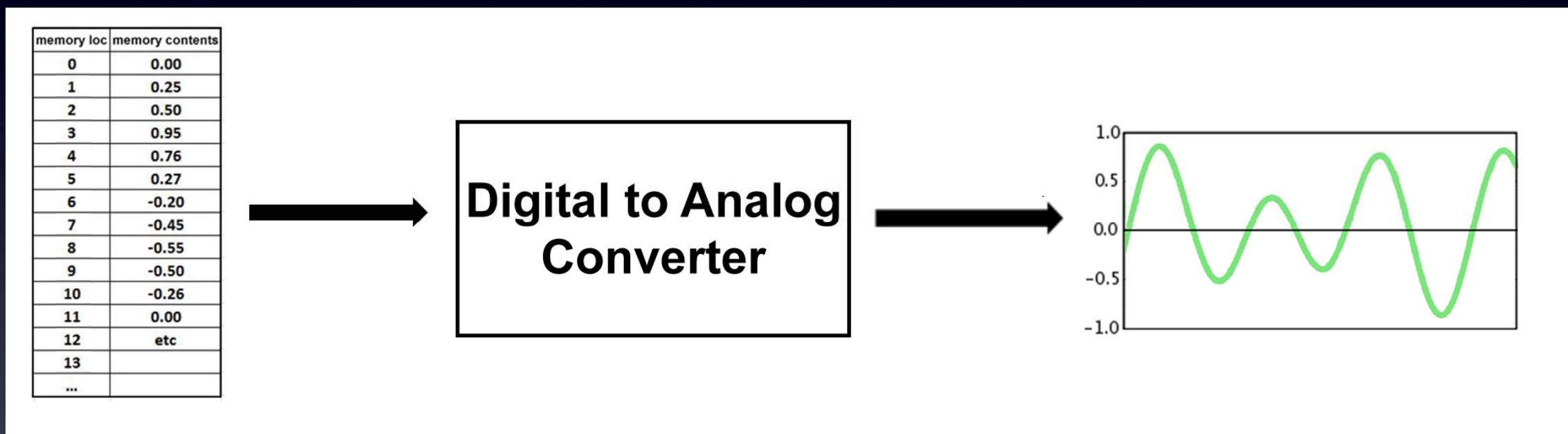
A/D

sampling an analog waveform to store it in digital memory

Digital Signal Processing

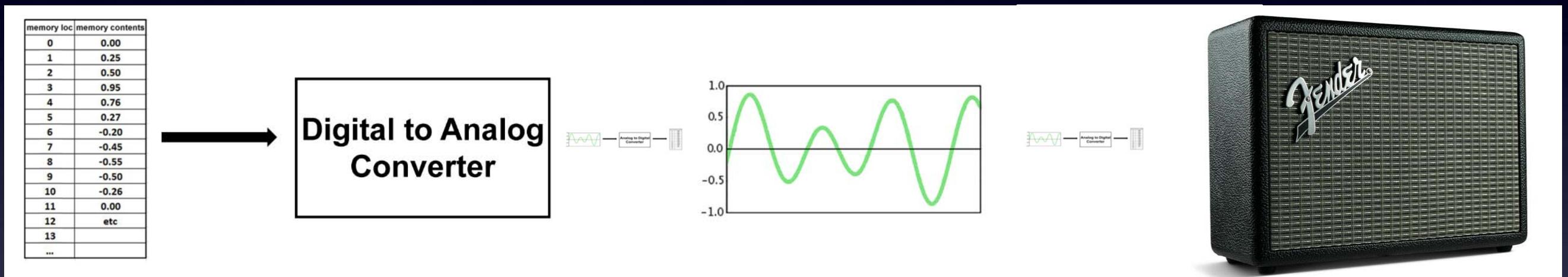
How do we play back a digitized waveform?

Digital Signal Processing



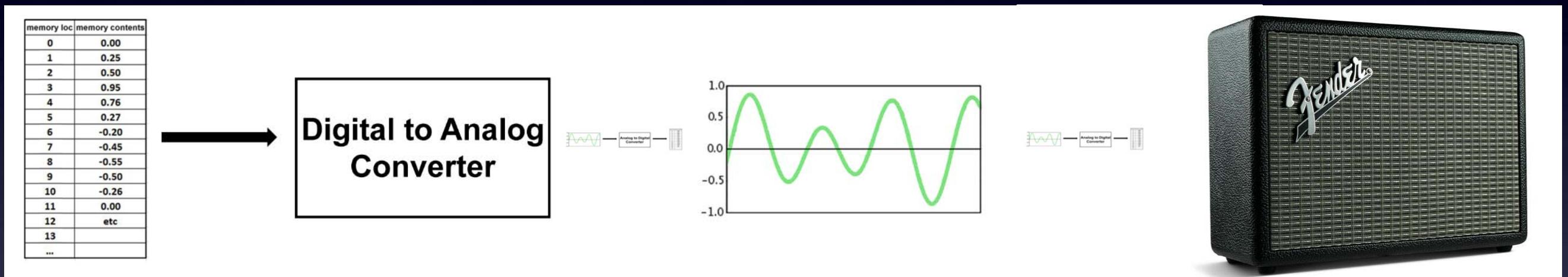
Digital to Analog Conversion:
Playing back the Digitized waveform

Digital Signal Processing



Digital to Analog Conversion:
Playing back the Digitized waveform

Digital Signal Processing



D/A

Playing back the Digitized waveform

Digital Signal Processing

How do you do
D/A ?

Digital Signal Processing

D/A chip (expensive)

or

PWM

PWM?

Digital Signal Processing



Square Wave:
ON half the time / OFF half of the time

PWM?

Digital Signal Processing



Square Wave:
ON half the time / OFF half of the time
(half the energy of ON all the time)

PWM?

Digital Signal Processing

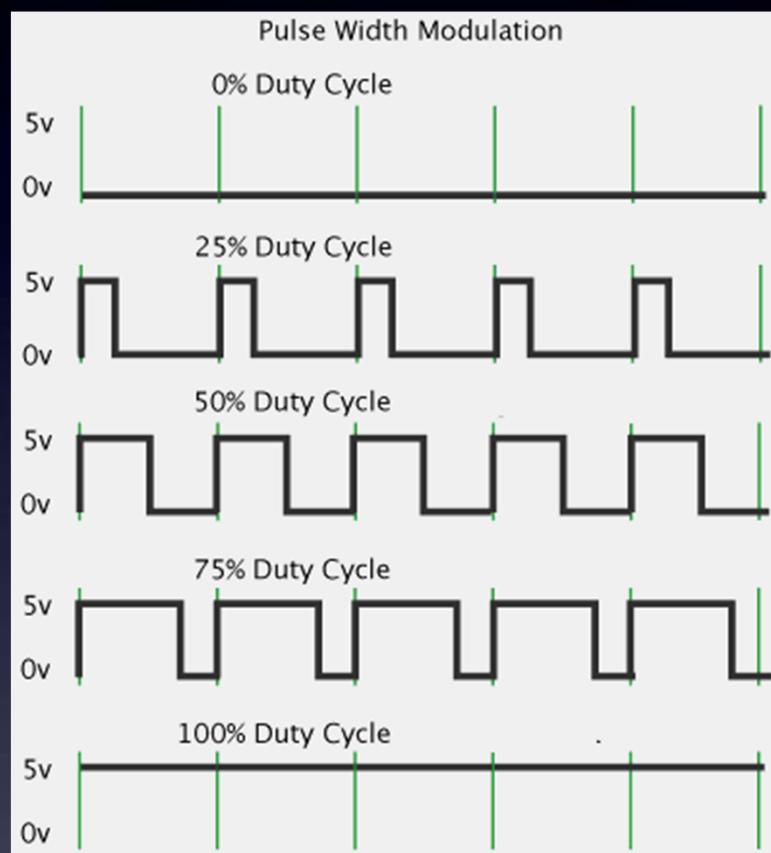


Pulse Wave:
ON and OFF at any ratio you like

This waveform: ON for 25% of the time / OFF for 75% of the time

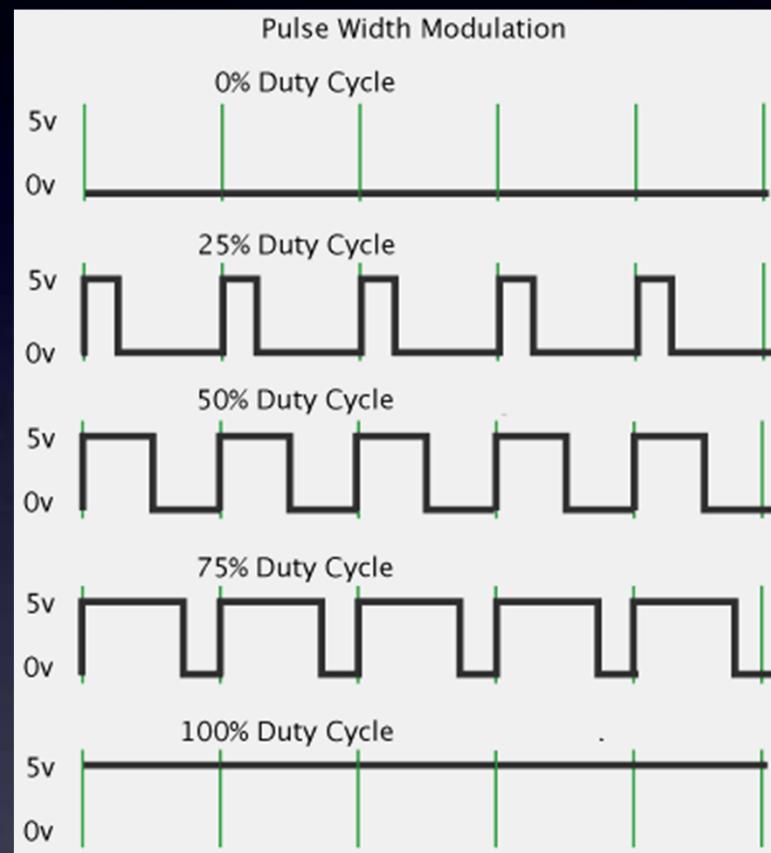
PWM?

Digital Signal Processing



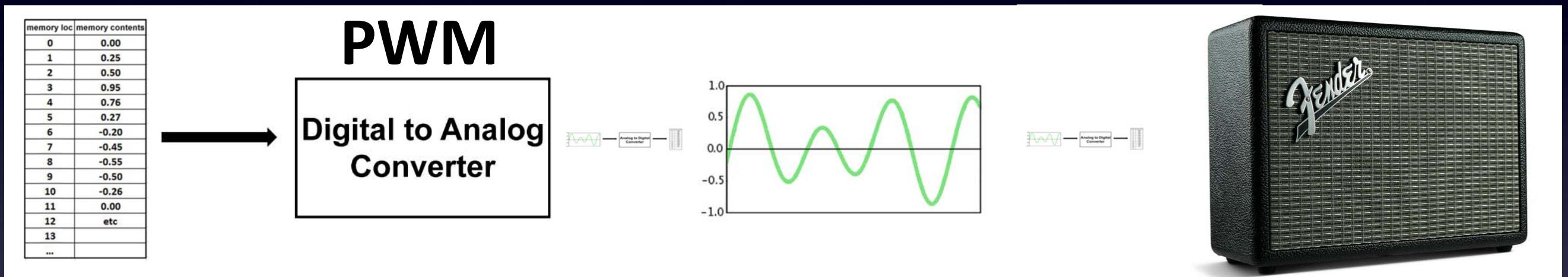
Pulse Wave:
ON and OFF at any ratio you like

Digital Signal Processing



PWM
Pulse Width Modulation

Digital Signal Processing



D/A

Using PWM for playing back the Digitized waveform

Digital Signal Processing

Kind of complicated to code

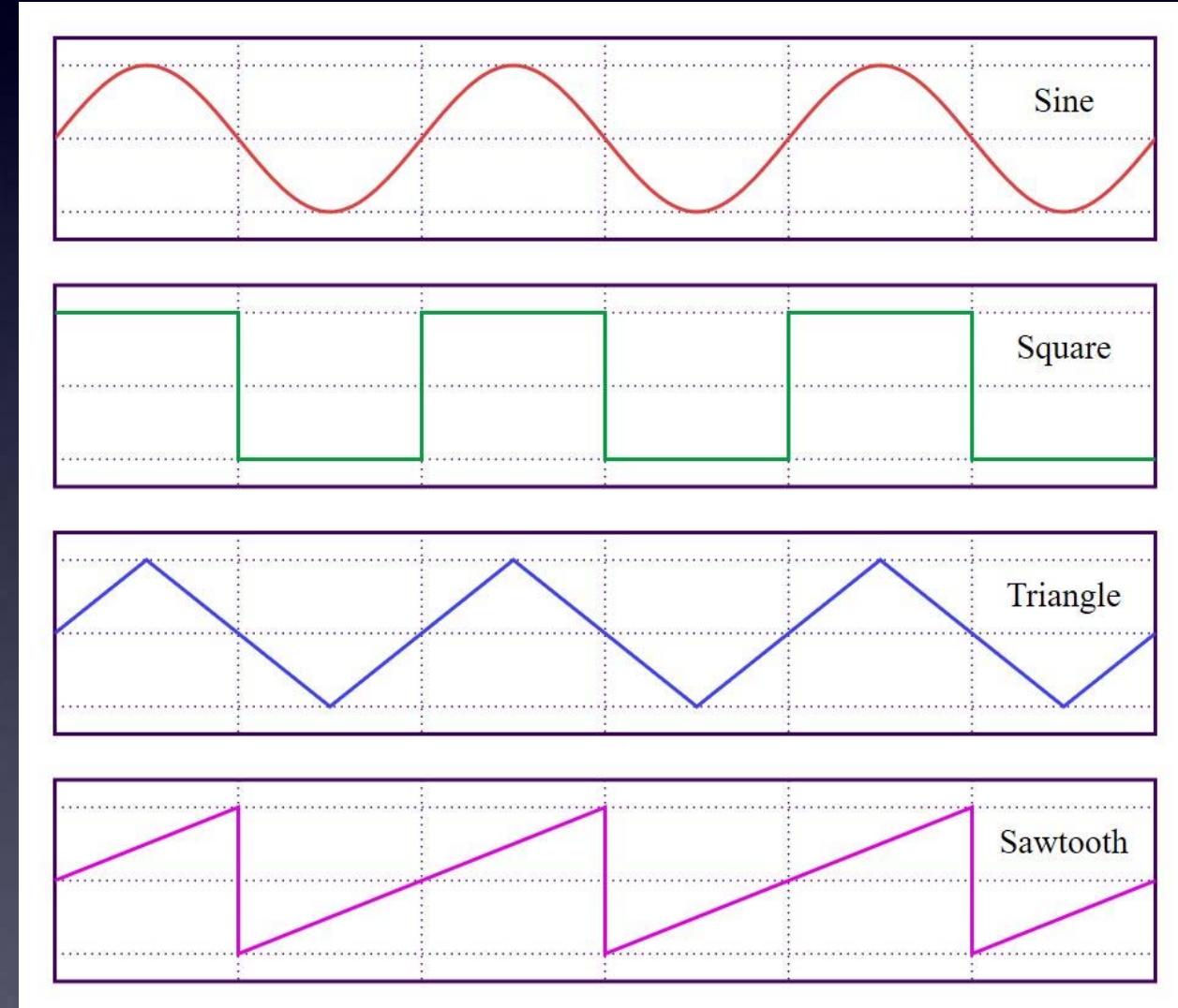
So, my ArduTouch software makes it easy

- Create “oscillators” with a couple lines of code
- Create “dynamics” with a couple lines of code

“Dynamics”
make the sound interesting

Digital Signal Processing

Some “Oscillators”:



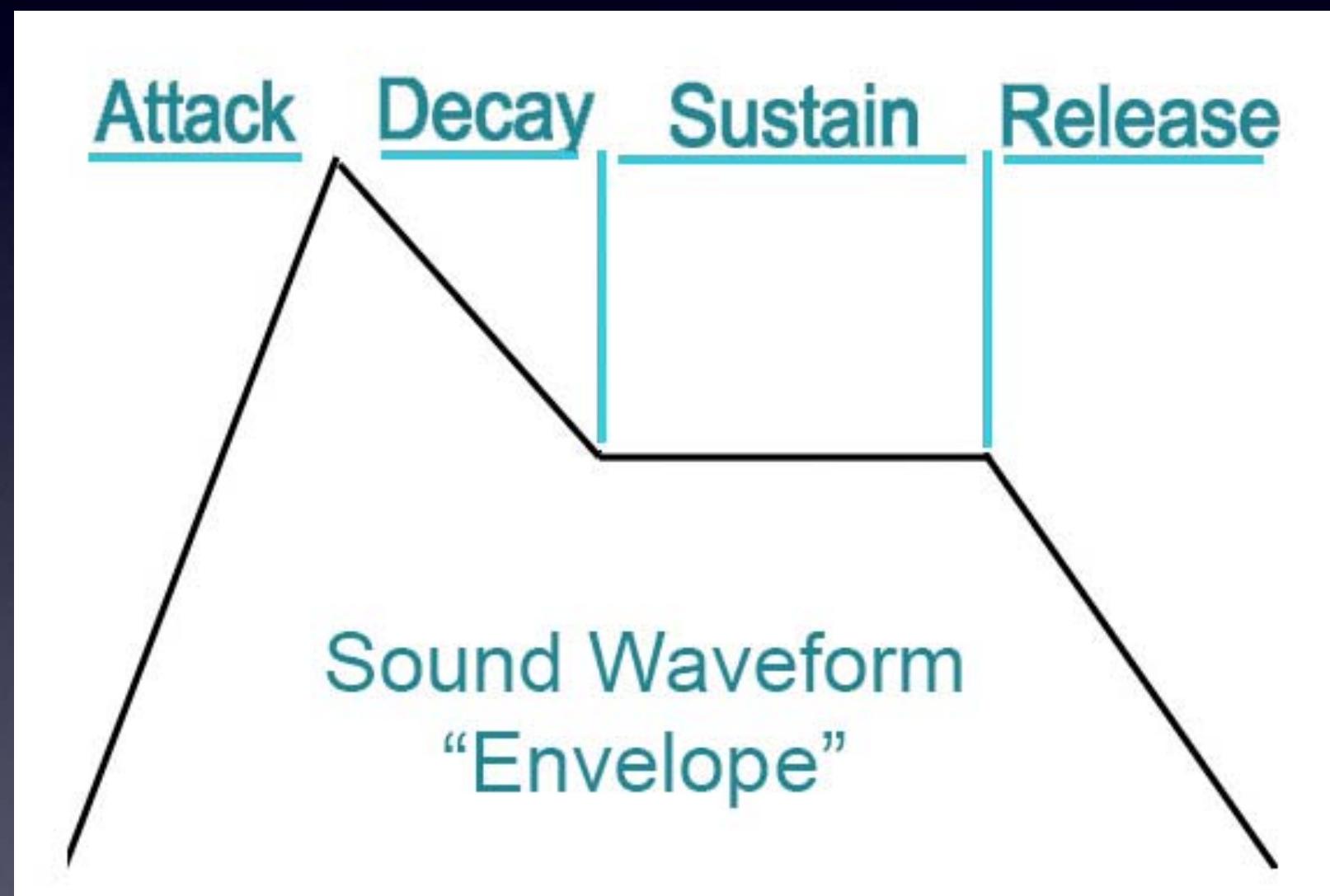
Digital Signal Processing

Some “Dynamics”:

- ADSR
- Tremolo
- Portamento
- Envelopes
- Filters
- Effects

Digital Signal Processing

ADSR:



Digital Signal Processing

Some “Dynamics”:

- ADSR
- Tremolo – *constant changing volume*
- Portamento
- Envelopes
- Filters
- Effects

Digital Signal Processing

Some “Dynamics”:

- ADSR
- Tremolo – *constant changing volume*
- Portamento – *glide between notes*
- Envelopes
- Filters
- Effects

Digital Signal Processing

Some “Dynamics”:

- ADSR
- Tremolo – *constant changing volume*
- Portamento – *glide between notes*
- Envelopes – *beyond ADSR*
- Filters
- Effects

Digital Signal Processing

Some “Dynamics”:

- ADSR
- Tremolo – *constant changing volume*
- Portamento – *glide between notes*
- Envelopes – *beyond ADSR*
- Filters – *like bass & treble – subtle to crazy*
- Effects

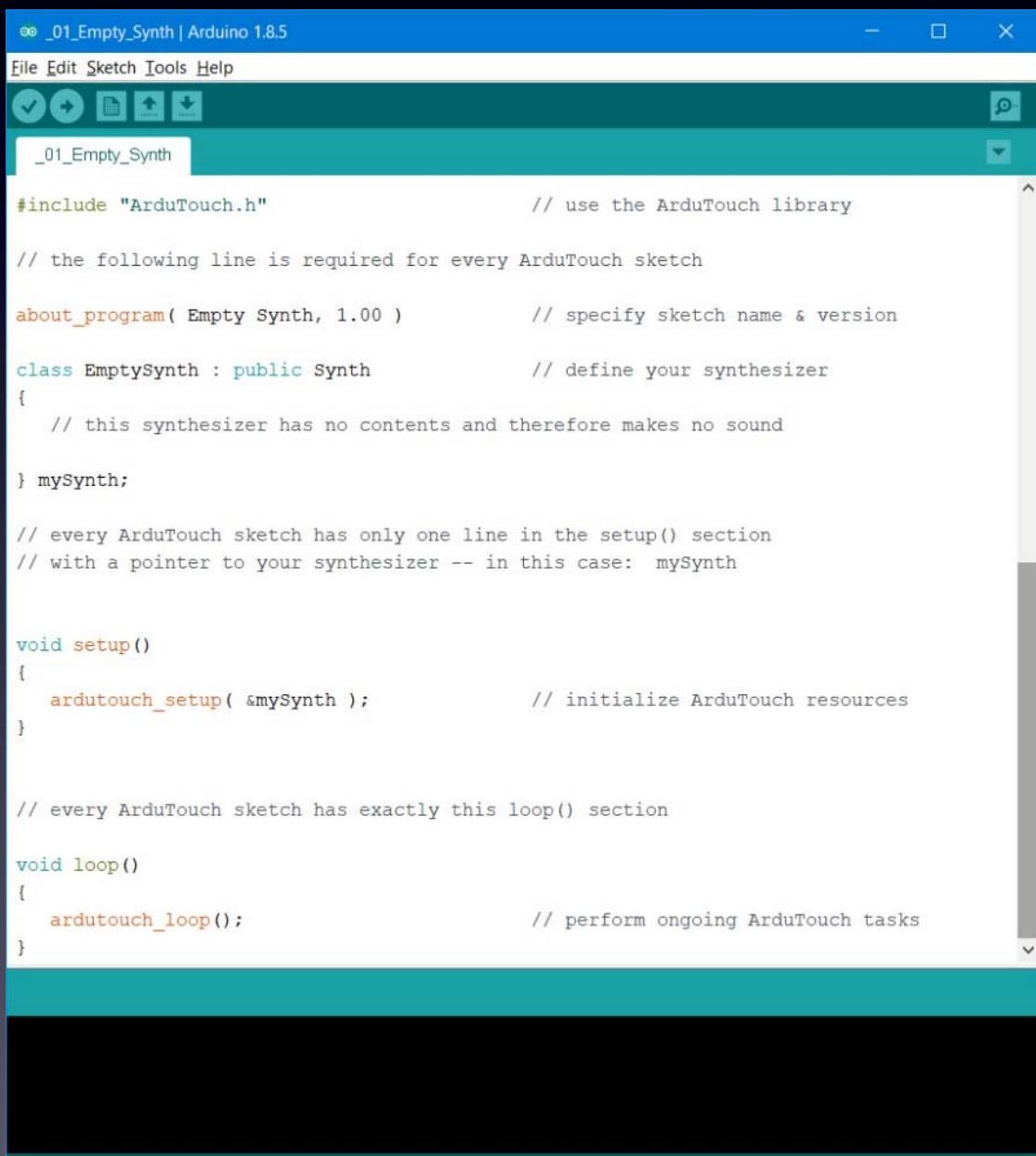
Digital Signal Processing

Some “Dynamics”:

- ADSR
- Tremolo – *constant changing volume*
- Portamento – *glide between notes*
- Envelopes – *beyond ADSR*
- Filters – *like bass & treble – subtle to crazy*
- Effects – *mess with the sound!*

ArduTouch

Arduino-Compatible



The screenshot shows the Arduino IDE interface with a sketch named '_01_Empty_Synth'. The code is as follows:

```
#include "ArduTouch.h" // use the ArduTouch library

// the following line is required for every ArduTouch sketch

about_program( Empty_Synth, 1.00 ) // specify sketch name & version

class EmptySynth : public Synth // define your synthesizer
{
    // this synthesizer has no contents and therefore makes no sound
} mySynth;

// every ArduTouch sketch has only one line in the setup() section
// with a pointer to your synthesizer -- in this case: mySynth

void setup()
{
    ardutouch_setup( &mySynth ); // initialize ArduTouch resources
}

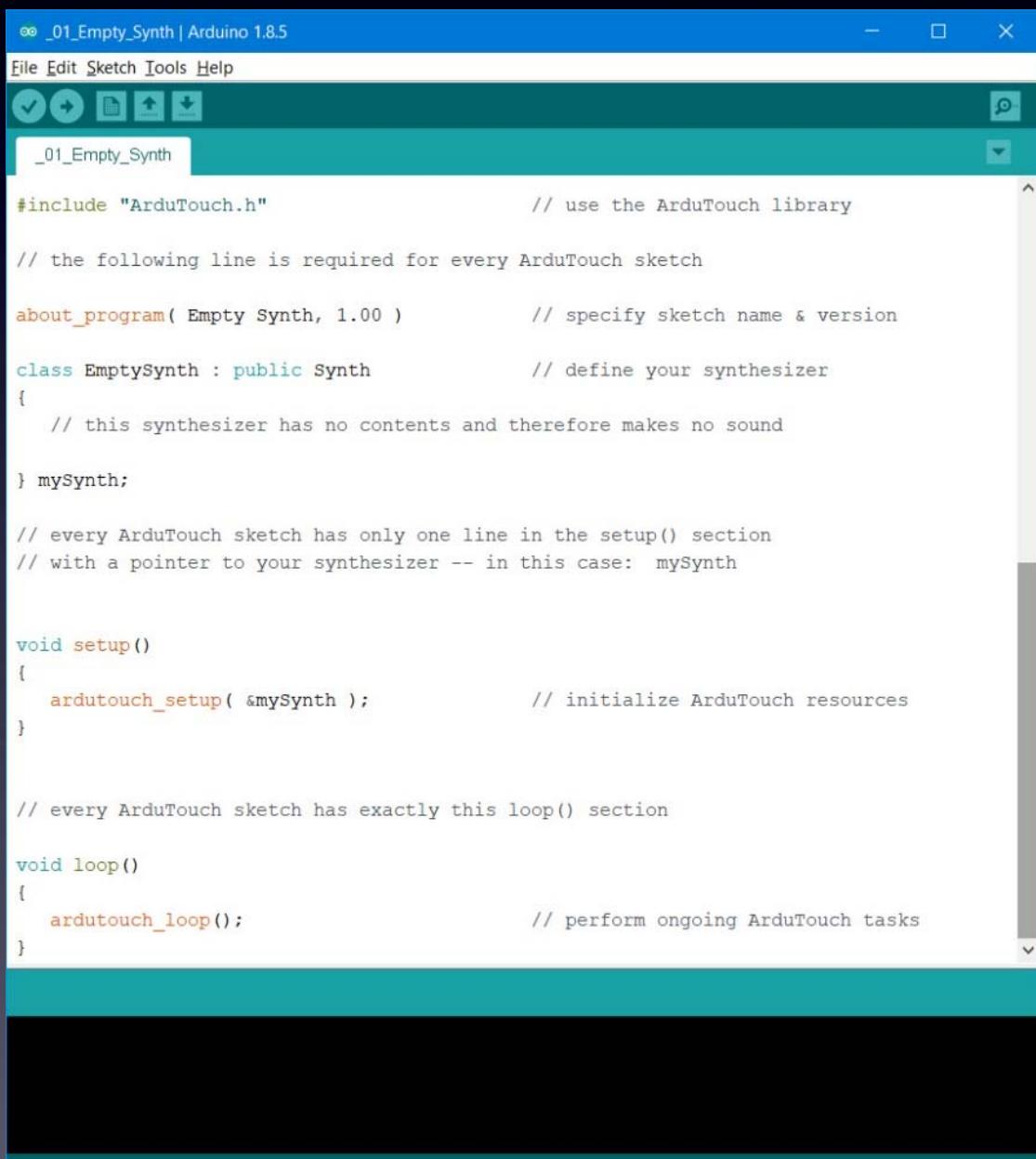
// every ArduTouch sketch has exactly this loop() section

void loop()
{
    ardutouch_loop(); // perform ongoing ArduTouch tasks
}
```

With
Tutorial examples

ArduTouch

Arduino-Compatible



The screenshot shows the Arduino IDE interface with a sketch named '_01_Empty_Synth'. The code is as follows:

```
#include "ArduTouch.h" // use the ArduTouch library

// the following line is required for every ArduTouch sketch

about_program( Empty_Synth, 1.00 ) // specify sketch name & version

class EmptySynth : public Synth // define your synthesizer
{
    // this synthesizer has no contents and therefore makes no sound
} mySynth;

// every ArduTouch sketch has only one line in the setup() section
// with a pointer to your synthesizer -- in this case: mySynth

void setup()
{
    ardutouch_setup( &mySynth ); // initialize ArduTouch resources
}

// every ArduTouch sketch has exactly this loop() section

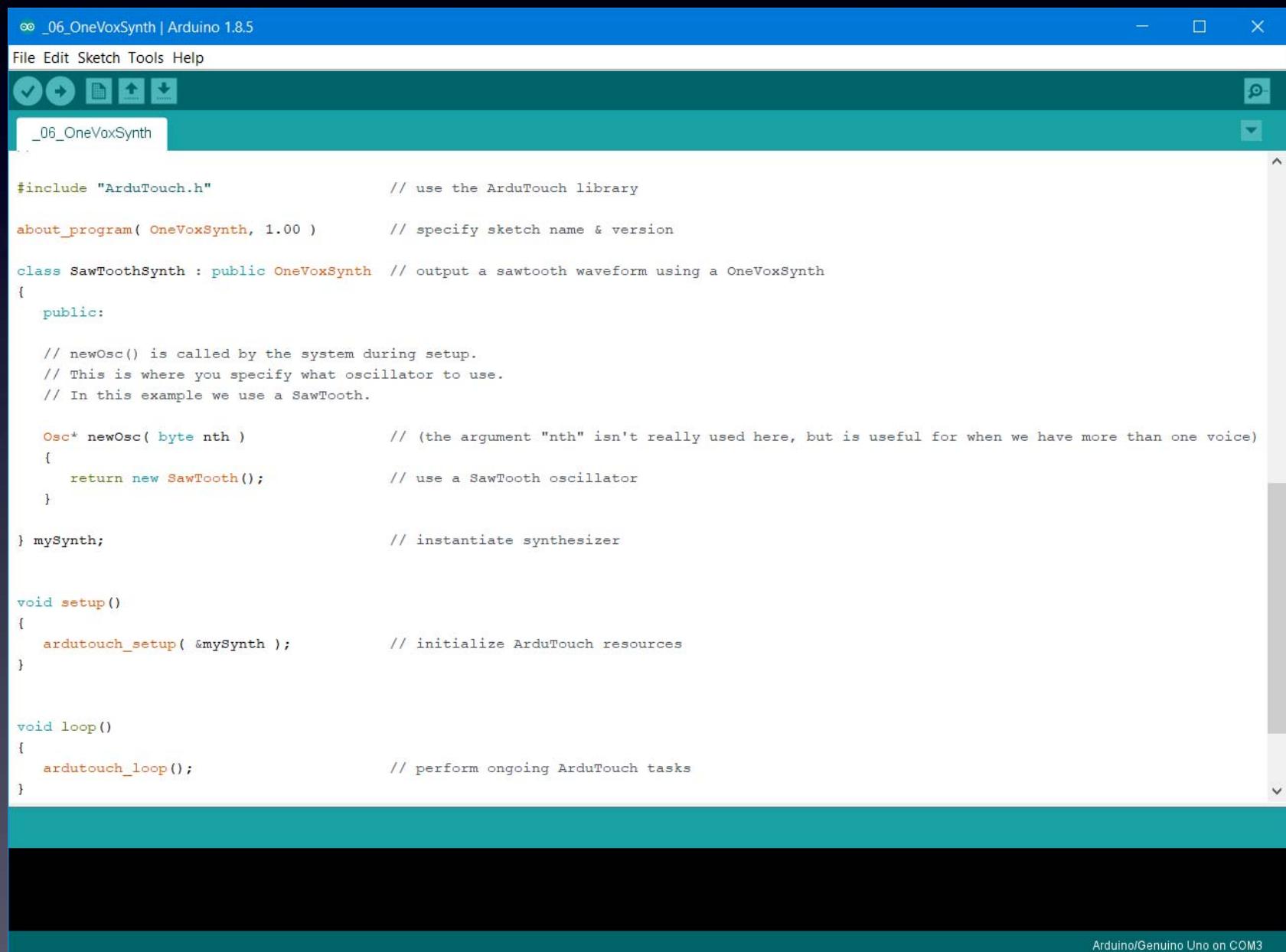
void loop()
{
    ardutouch_loop(); // perform ongoing ArduTouch tasks
}
```

With
Tutorial examples

*Follow examples
01 through 09
to easily learn
to code your own
synthesizers*

ArduTouch

Arduino-Compatible



The screenshot shows the Arduino IDE interface with a sketch titled '_06_OneVoxSynth'. The code uses the ArduTouch library to create a synthesizer. It includes setup and loop functions, and defines a class SawToothSynth that outputs a sawtooth waveform.

```
#include "ArduTouch.h" // use the ArduTouch library

about_program( OneVoxSynth, 1.00 ) // specify sketch name & version

class SawToothSynth : public OneVoxSynth // output a sawtooth waveform using a OneVoxSynth
{
public:

    // newOsc() is called by the system during setup.
    // This is where you specify what oscillator to use.
    // In this example we use a SawTooth.

    Osc* newOsc( byte nth ) // (the argument "nth" isn't really used here, but is useful for when we have more than one voice)
    {
        return new SawTooth(); // use a SawTooth oscillator
    }

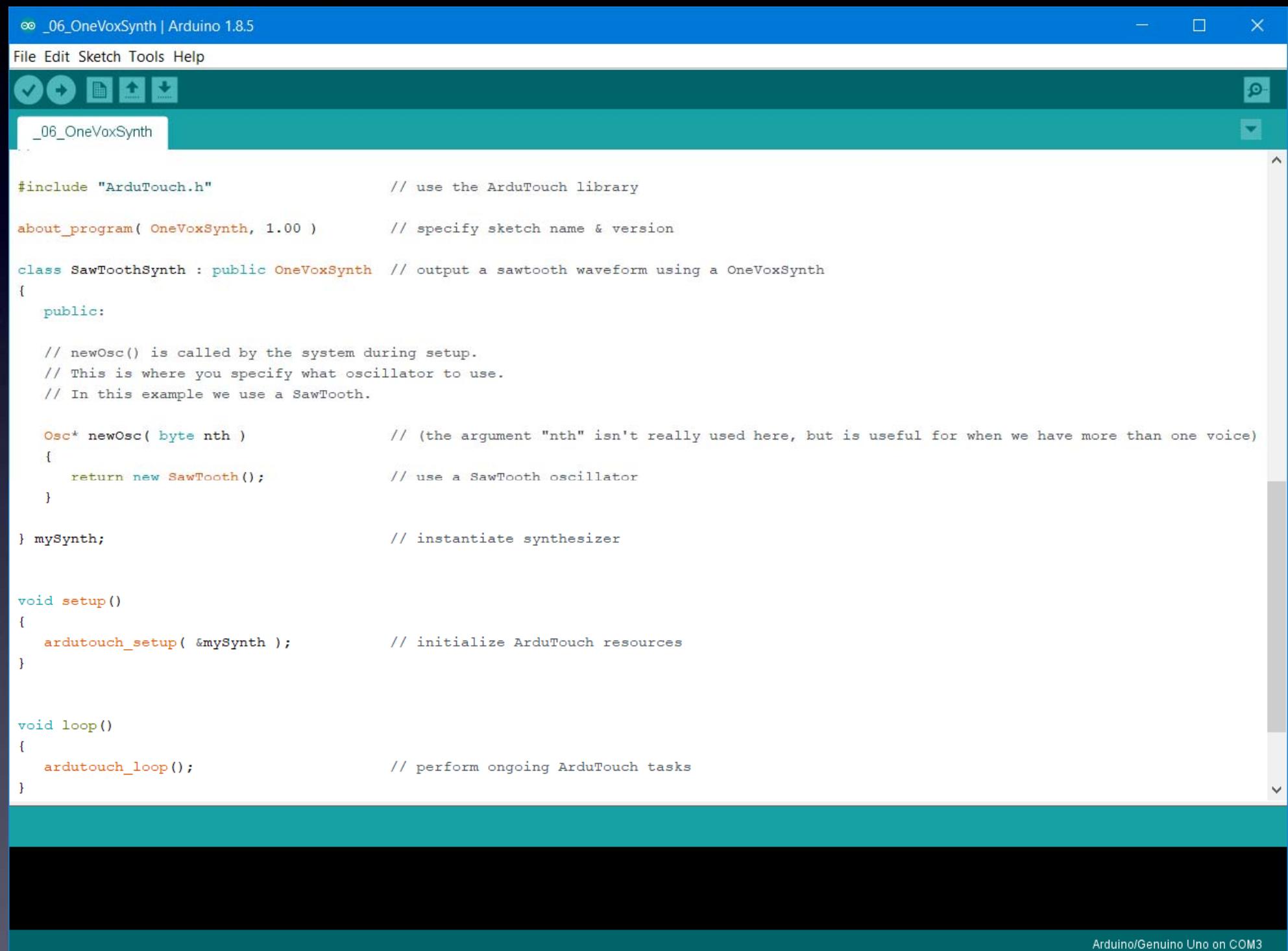
} mySynth; // instantiate synthesizer

void setup()
{
    ardutouch_setup( &mySynth ); // initialize ArduTouch resources
}

void loop()
{
    ardutouch_loop(); // perform ongoing ArduTouch tasks
}
```

With extensive
Arduino library
for ArduTouch
*to make it easy
to create
your own synths*

ArduTouch



The screenshot shows the Arduino IDE interface with the sketch `_06_OneVoxSynth` open. The code implements a synthesizer using the ArduTouch library. It includes a class `SawToothSynth` that outputs a sawtooth waveform. The sketch initializes the ArduTouch resources in `setup()` and performs ongoing tasks in `loop()`.

```
#include "ArduTouch.h"          // use the ArduTouch library

about_program( OneVoxSynth, 1.00 )      // specify sketch name & version

class SawToothSynth : public OneVoxSynth // output a sawtooth waveform using a OneVoxSynth
{
public:

  // newOsc() is called by the system during setup.
  // This is where you specify what oscillator to use.
  // In this example we use a SawTooth.

  Osc* newOsc( byte nth )           // (the argument "nth" isn't really used here, but is useful for when we have more than one voice)
  {
    return new SawTooth();           // use a SawTooth oscillator
  }

} mySynth;                         // instantiate synthesizer

void setup()
{
  ardutouch_setup( &mySynth );      // initialize ArduTouch resources
}

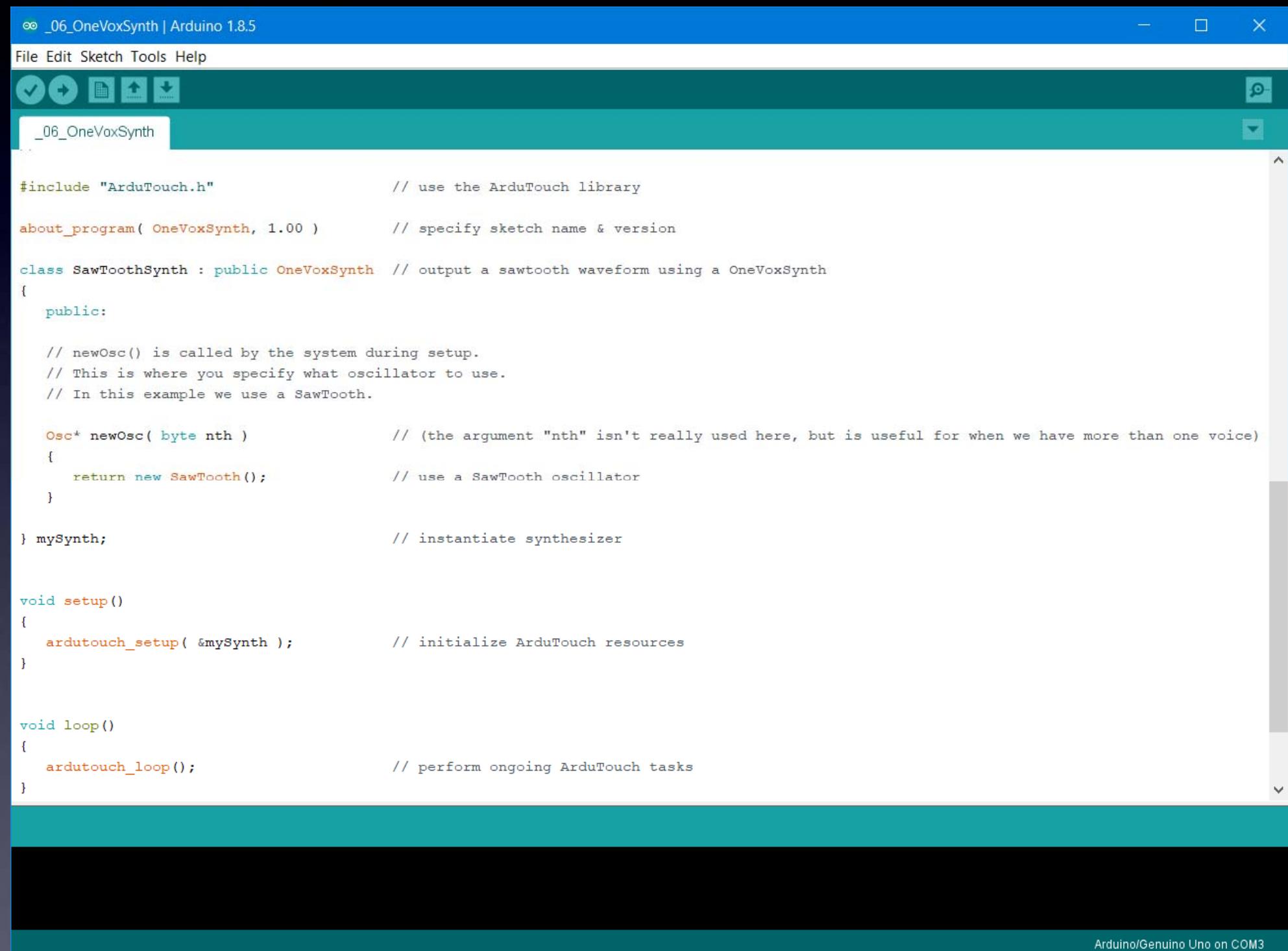
void loop()
{
  ardutouch_loop();                // perform ongoing ArduTouch tasks
}
```

Arduino/Genuino Uno on COM3

Complete code for:

- sawtooth waves
- play with keyboard
- change octaves
- volume control

ArduTouch



The screenshot shows the Arduino IDE interface with the sketch `_06_OneVoxSynth` open. The code implements a simple synthesizer using the ArduTouch library. It includes setup for the library, creation of a `SawToothSynth` object, and a loop that performs ongoing tasks.

```
#include "ArduTouch.h"          // use the ArduTouch library

about_program( OneVoxSynth, 1.00 )    // specify sketch name & version

class SawToothSynth : public OneVoxSynth // output a sawtooth waveform using a OneVoxSynth
{
public:
  // newOsc() is called by the system during setup.
  // This is where you specify what oscillator to use.
  // In this example we use a SawTooth.

  Osc* newOsc( byte nth )        // (the argument "nth" isn't really used here, but is useful for when we have more than one voice)
  {
    return new SawTooth();        // use a SawTooth oscillator
  }

} mySynth;                      // instantiate synthesizer

void setup()
{
  ardutouch_setup( &mySynth );    // initialize ArduTouch resources
}

void loop()
{
  ardutouch_loop();              // perform ongoing ArduTouch tasks
}
```

Arduino/Genuino Uno on COM3

Easy to add:

- Tremolo
- Portamento
- Envelopes
- Filters
- Effects
- Other waveforms



ArduTouch

The screenshot shows a GitHub repository page for 'maltman23 / ArduTouch'. The repository has 7 commits, 1 branch, 0 releases, and 1 contributor. The commit history lists several updates to Arduino files, assembly instructions, BOM, Eagle, Schematic, .gitattributes, .gitignore, and README.md, all made by 'maltman23' 7 months ago. The latest commit is 'b5109e5' on Jul 10.

File	Description	Time Ago
Arduino	update to PCB v1.2 rev C & DuoPoly v2.05	5 months ago
AssemblyInstructions	ArduTouch assembly instructions	7 months ago
BOM	ArduTouch BOM	7 months ago
Eagle	update to PCB v1.2 rev C & DuoPoly v2.05	5 months ago
Schematic	update to PCB v1.2 rev C & DuoPoly v2.05	5 months ago
.gitattributes	Added .gitattributes & .gitignore files	7 months ago
.gitignore	Added .gitattributes & .gitignore files	7 months ago
README.md	Create README.md	7 months ago

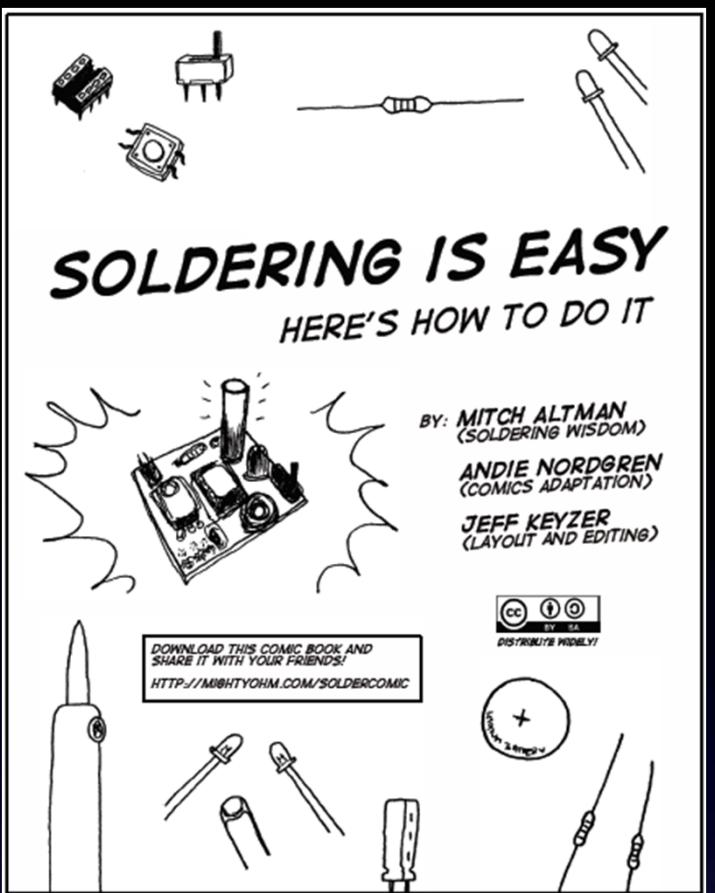
**Open Hardware – everything is on Github
maltman23**

(Don't bring these
home)

Tools



Learn To Solder



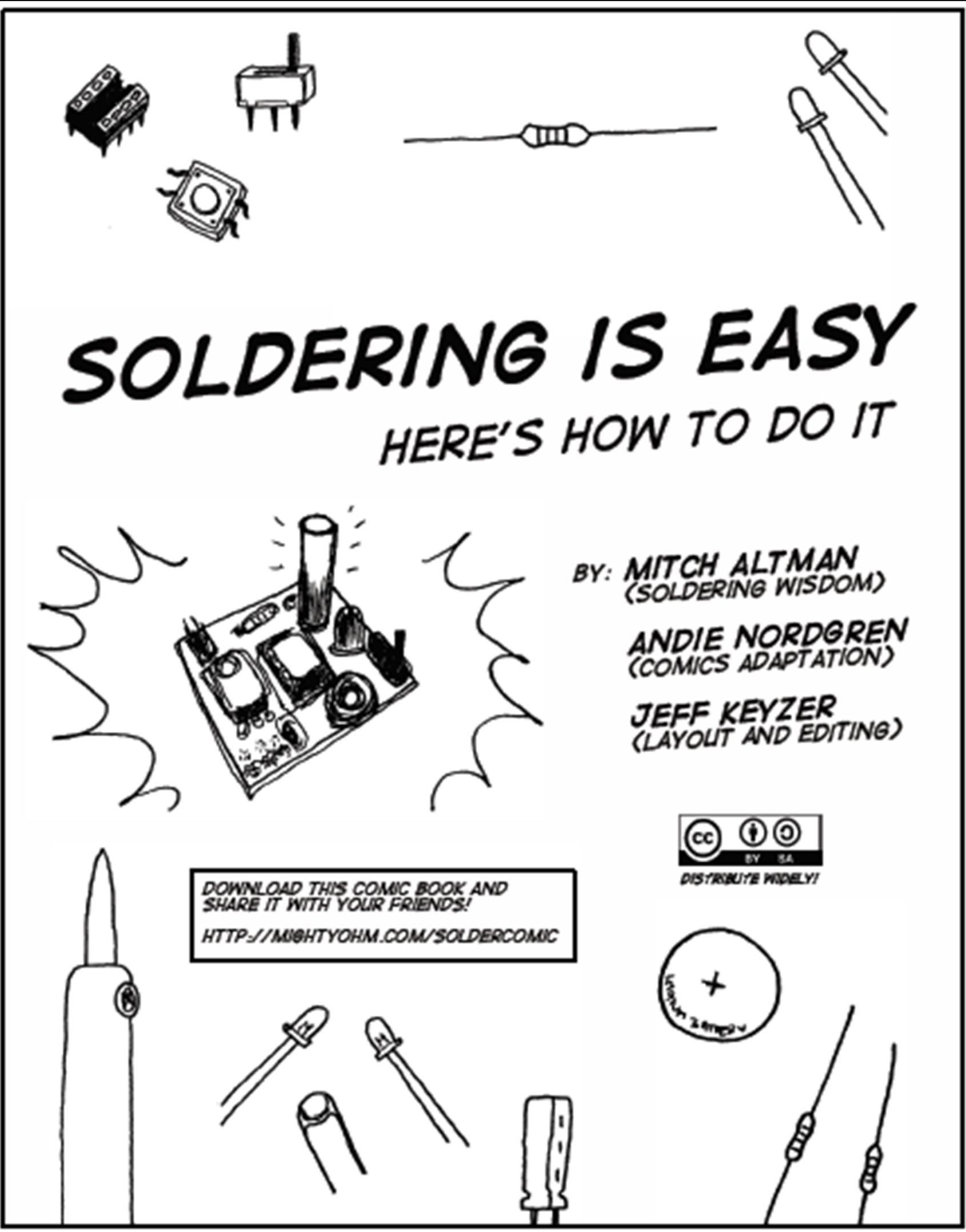
The following photos will show you how to solder.
But feel free to download the “Soldering Is Easy” comic book for free!

(In many different languages.)

download for free at:

<http://mightyohm.com/soldercomic>

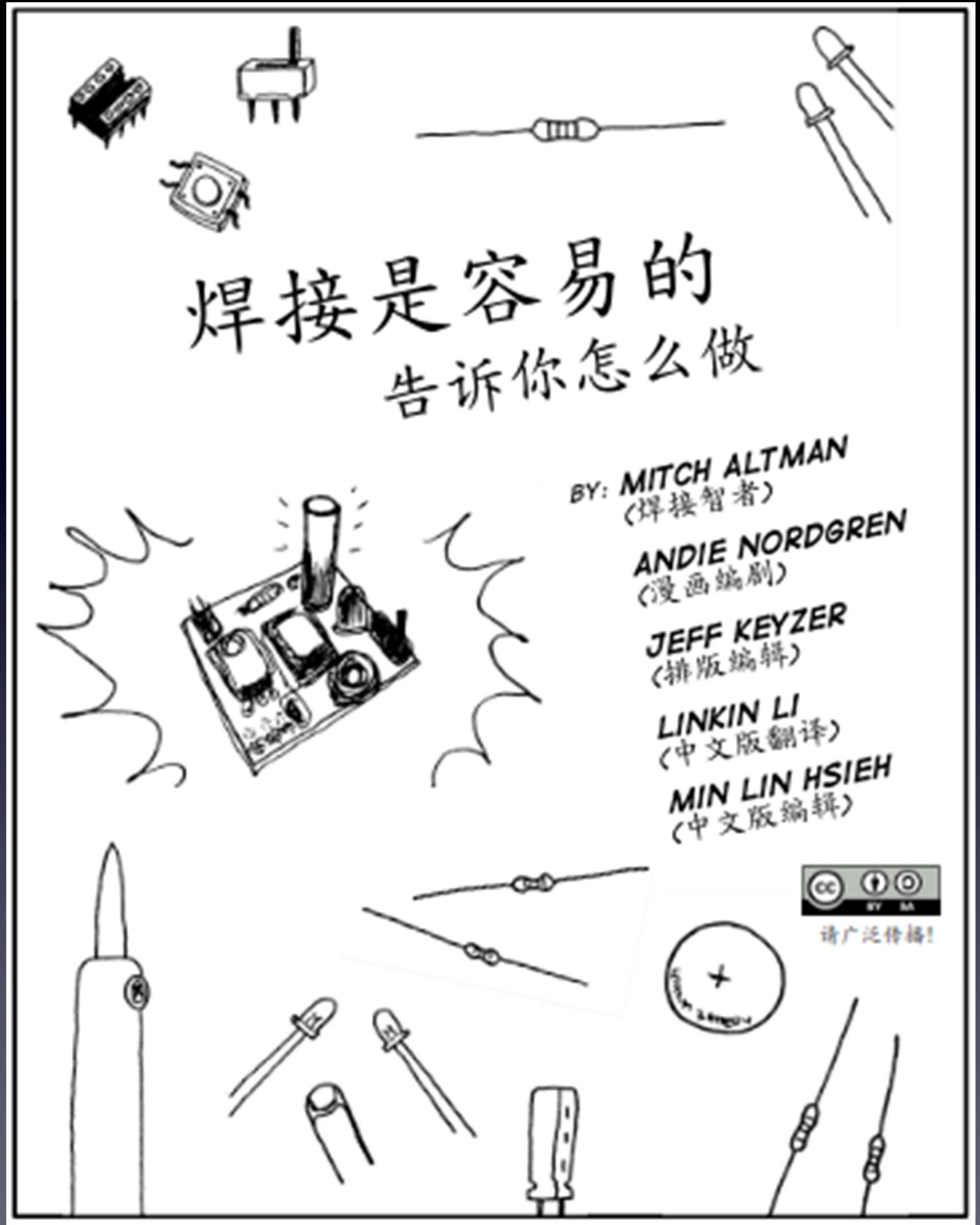
Learn To Solder



download for free at:

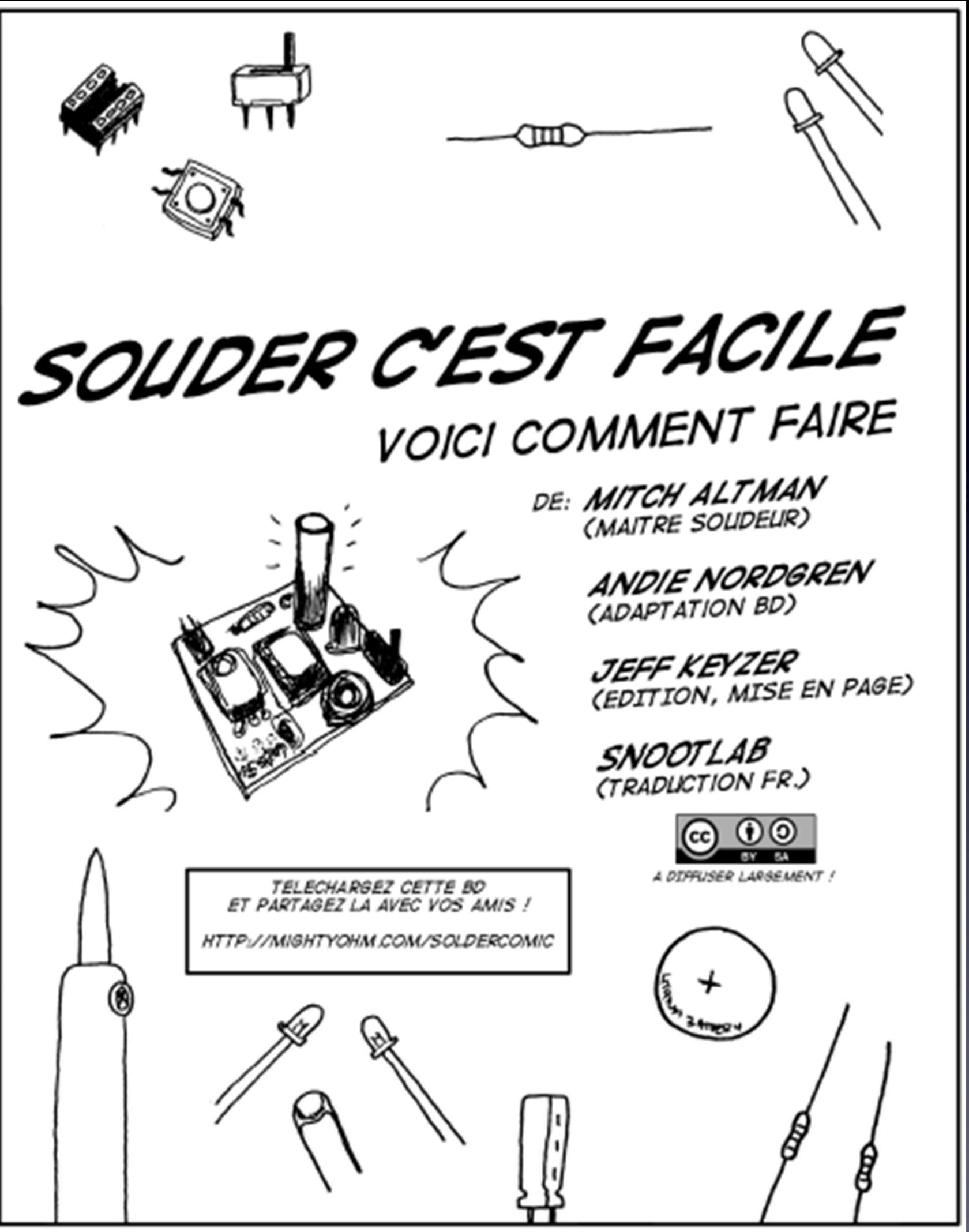
<http://mightyohm.com/soldercomic>

Learn To Solder



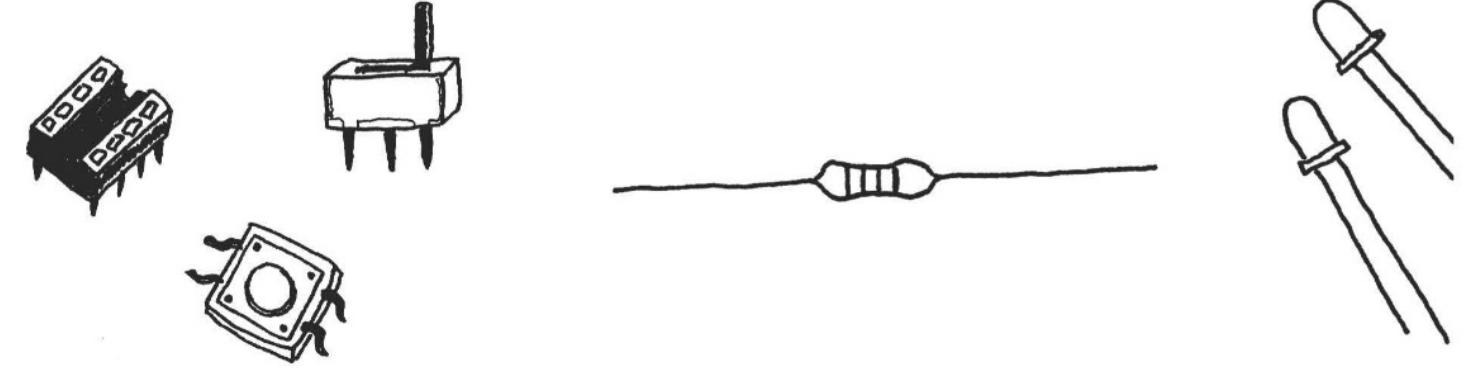
Download in the language of your choice for free at:
<http://mightyohm.com/soldercomic>

Learn To Solder



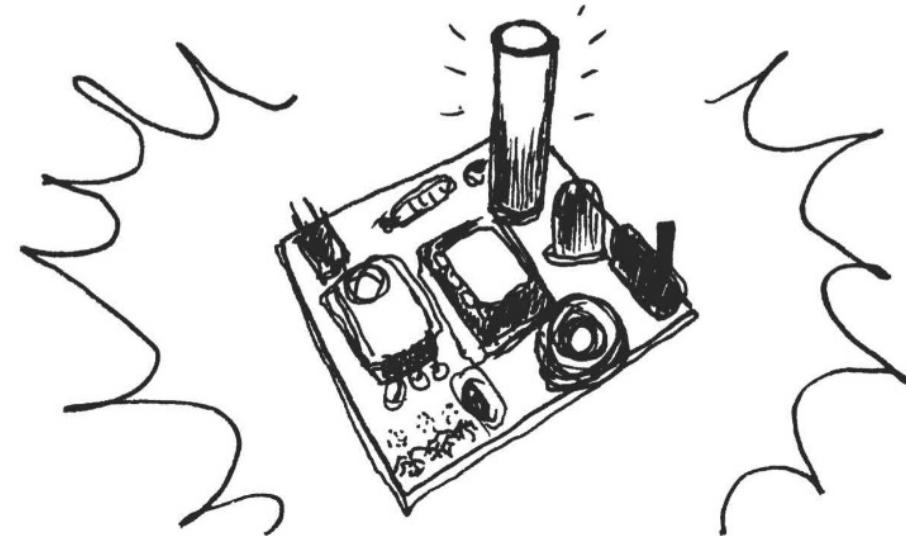
Download in the language of your choice for free at:
<http://mightyohm.com/soldercomic>

Learn To Solder



SOLDAR ES FÁCIL!

APRENDE CÓMO HACERLO



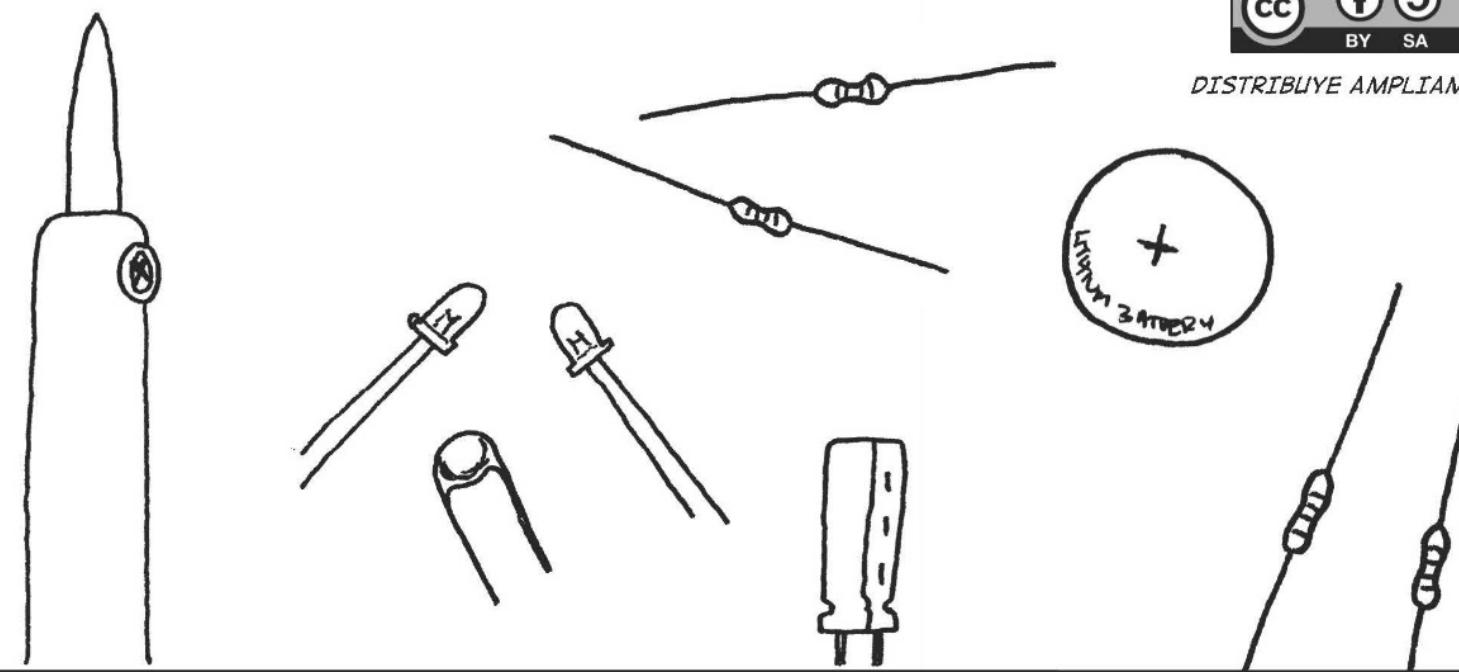
POR: MITCH ALTMAN
(SABIDURÍA EN SOLDADO)

ANDIE NORDGREN
(ADAPTACIÓN A COMIC)

JEFF KEYZER
(DISEÑO Y EDICIÓN)



DISTRIBUYE AMPLIAMENTE!



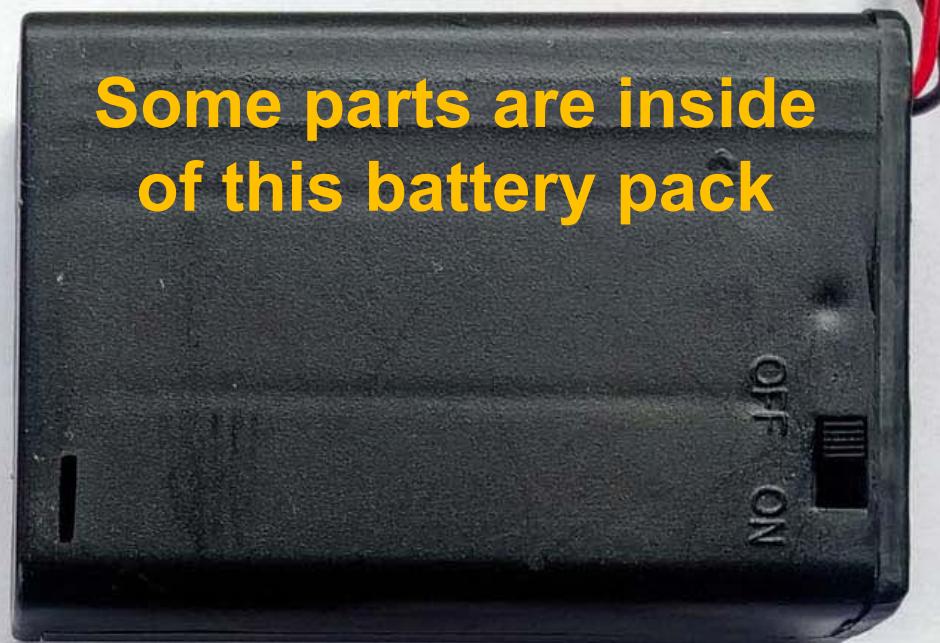
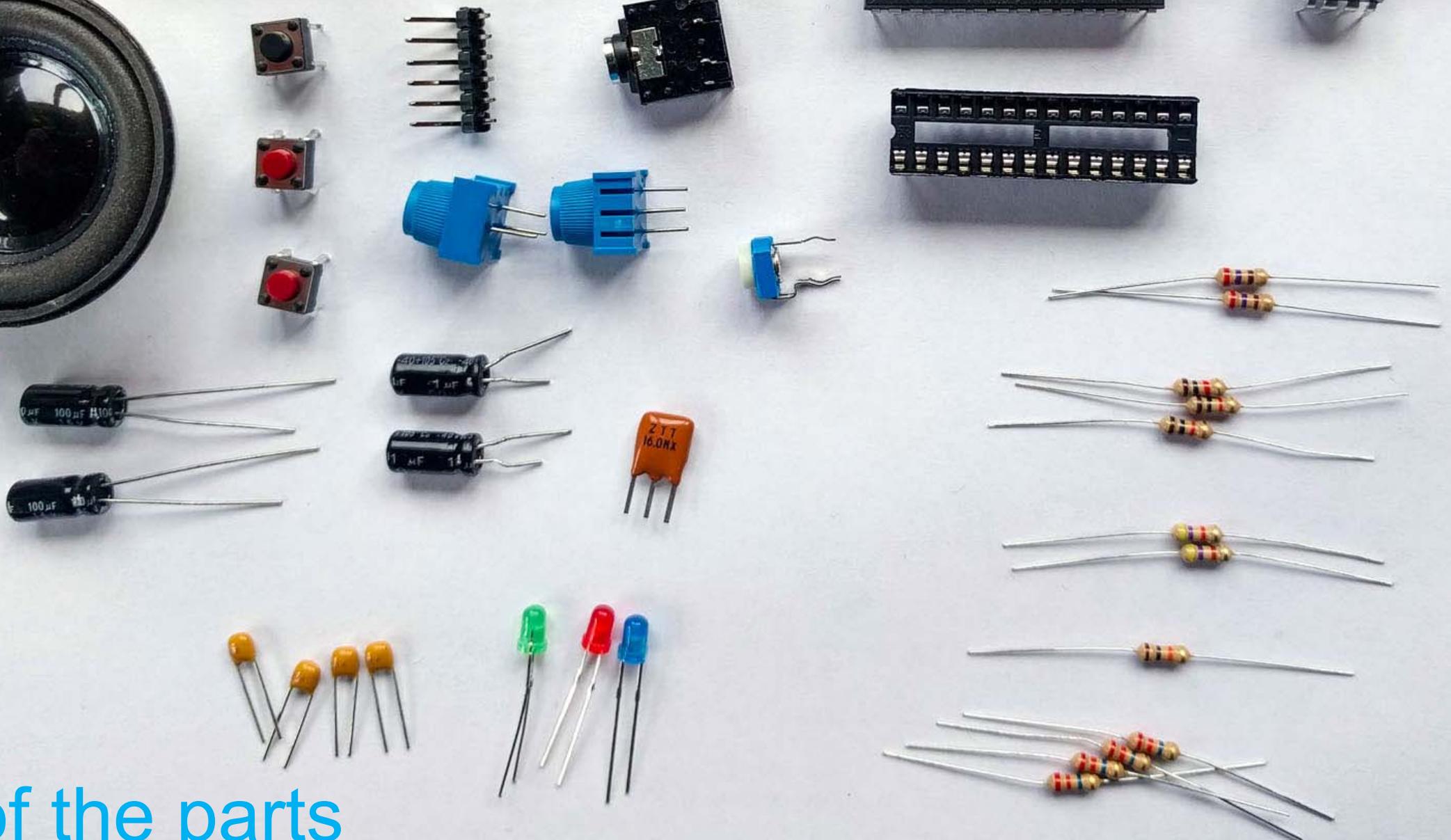
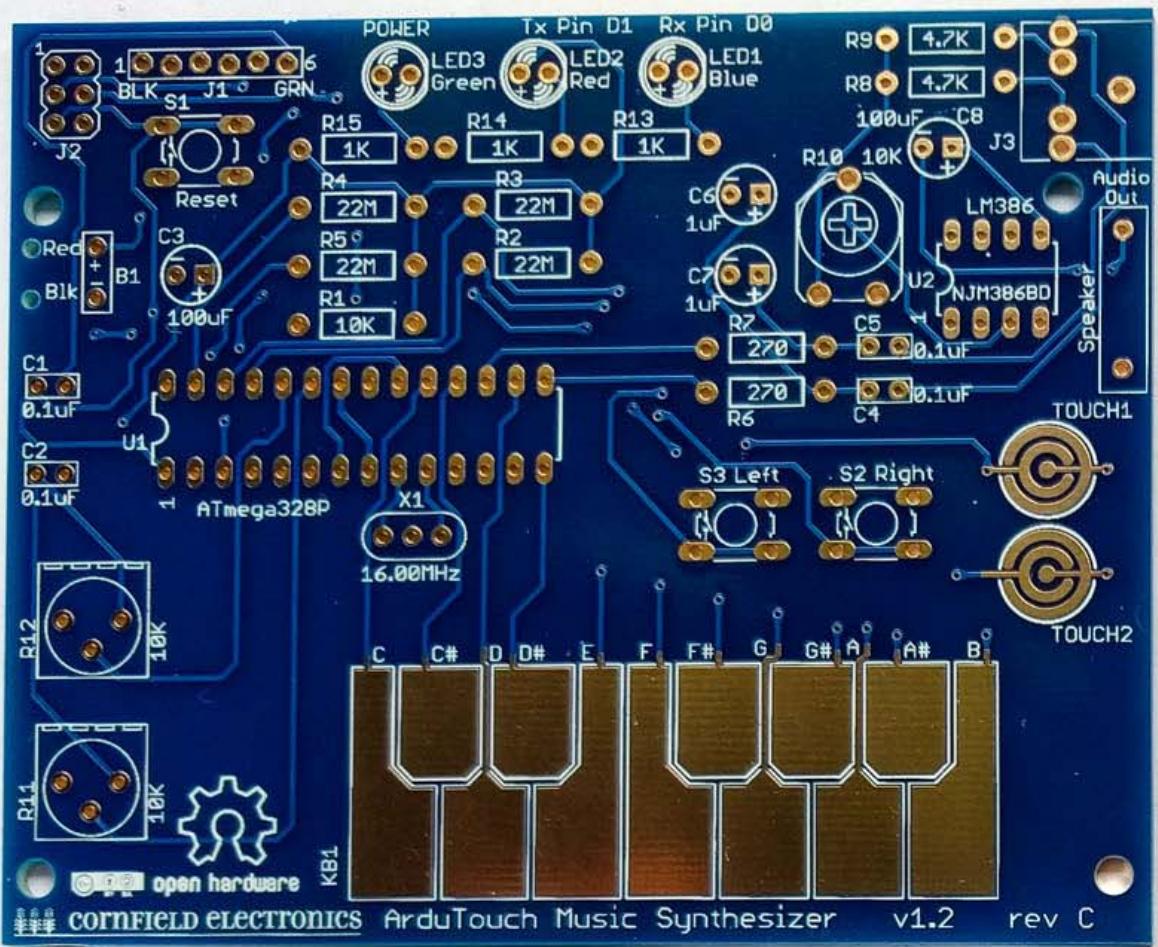
Download in the language of your choice for free at:

<http://mightyohm.com/soldercomic>

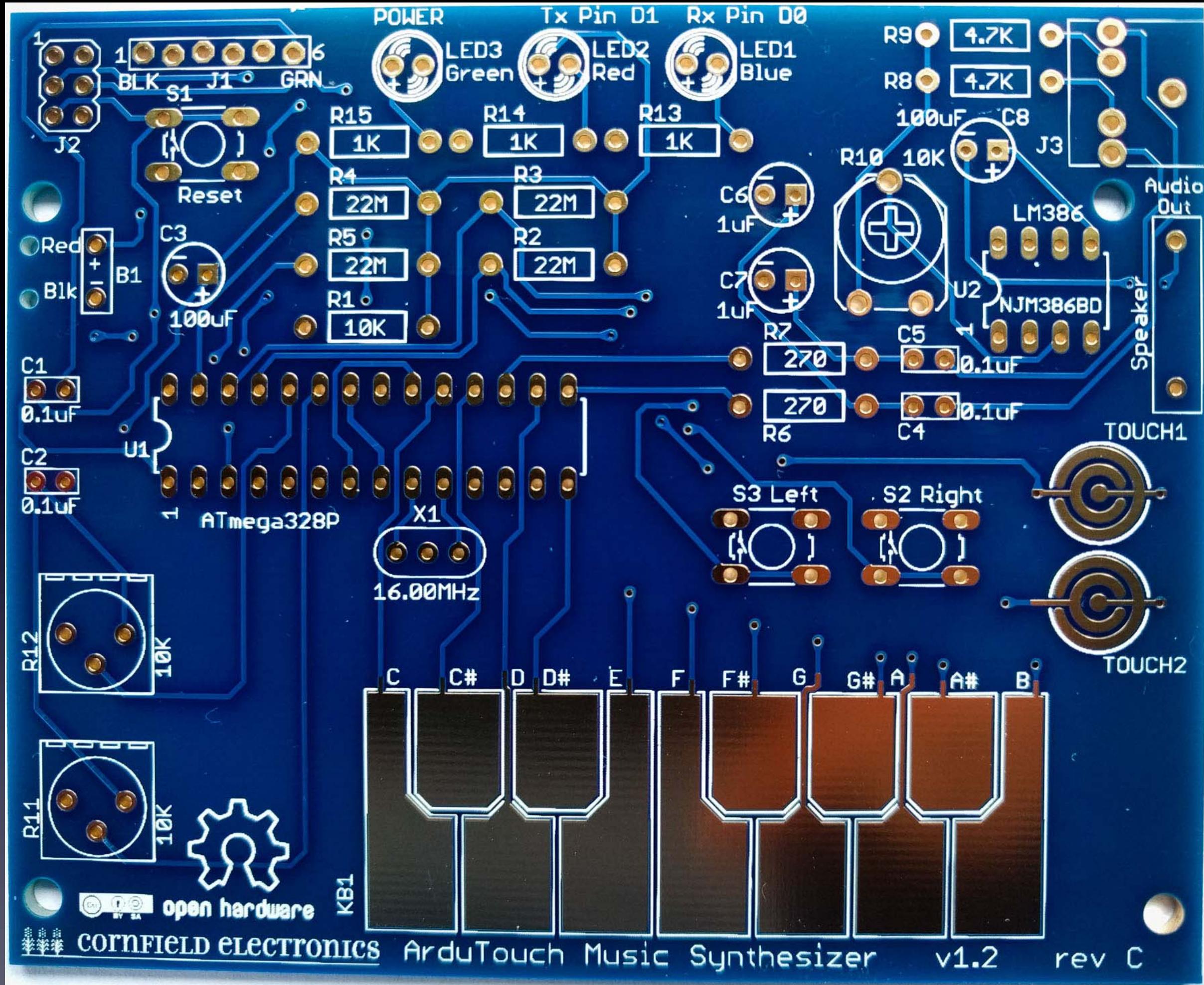
Learn To Solder

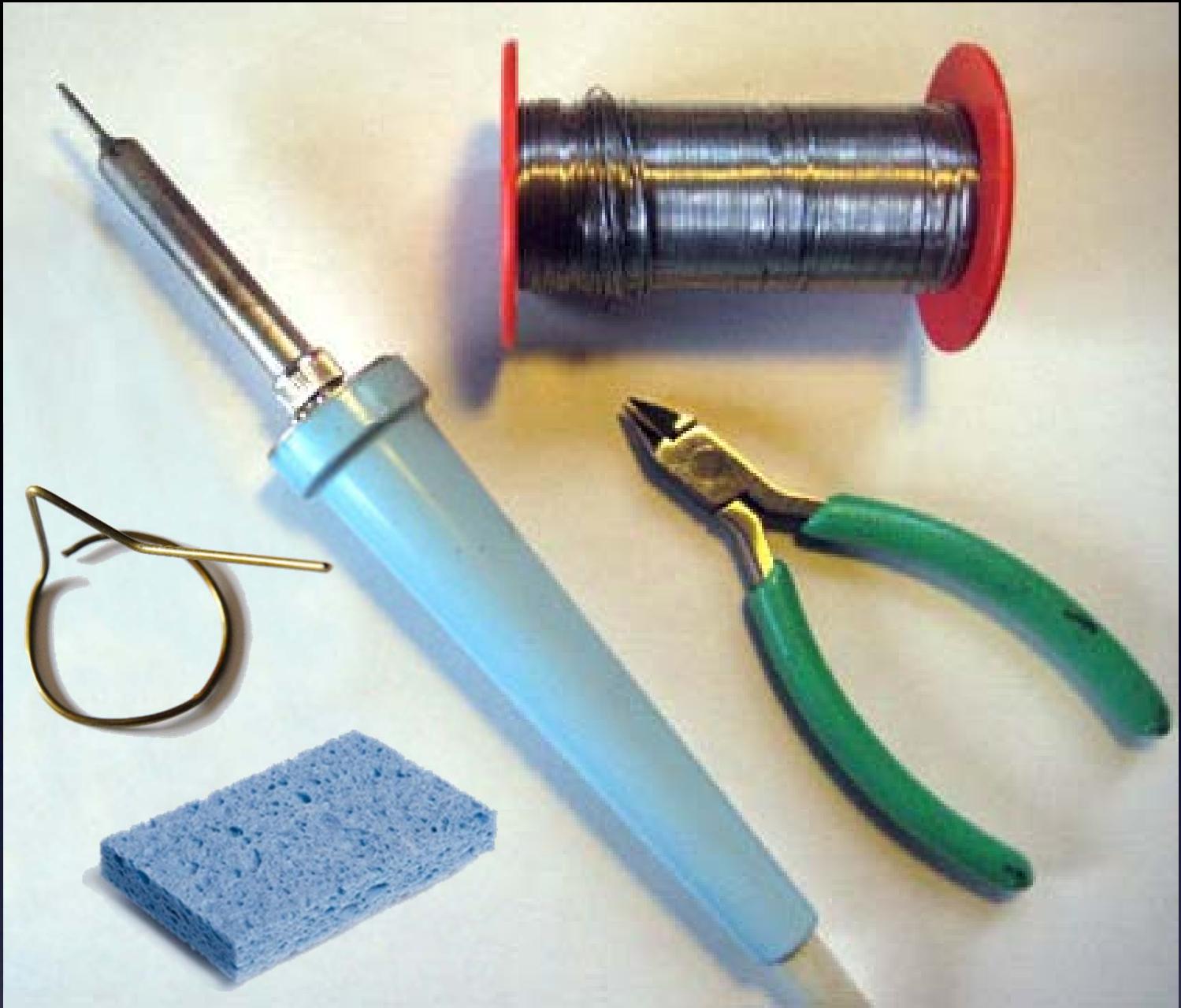


Download in the language of your choice for free at:
<http://mightyohm.com/soldercomic>



All of the parts



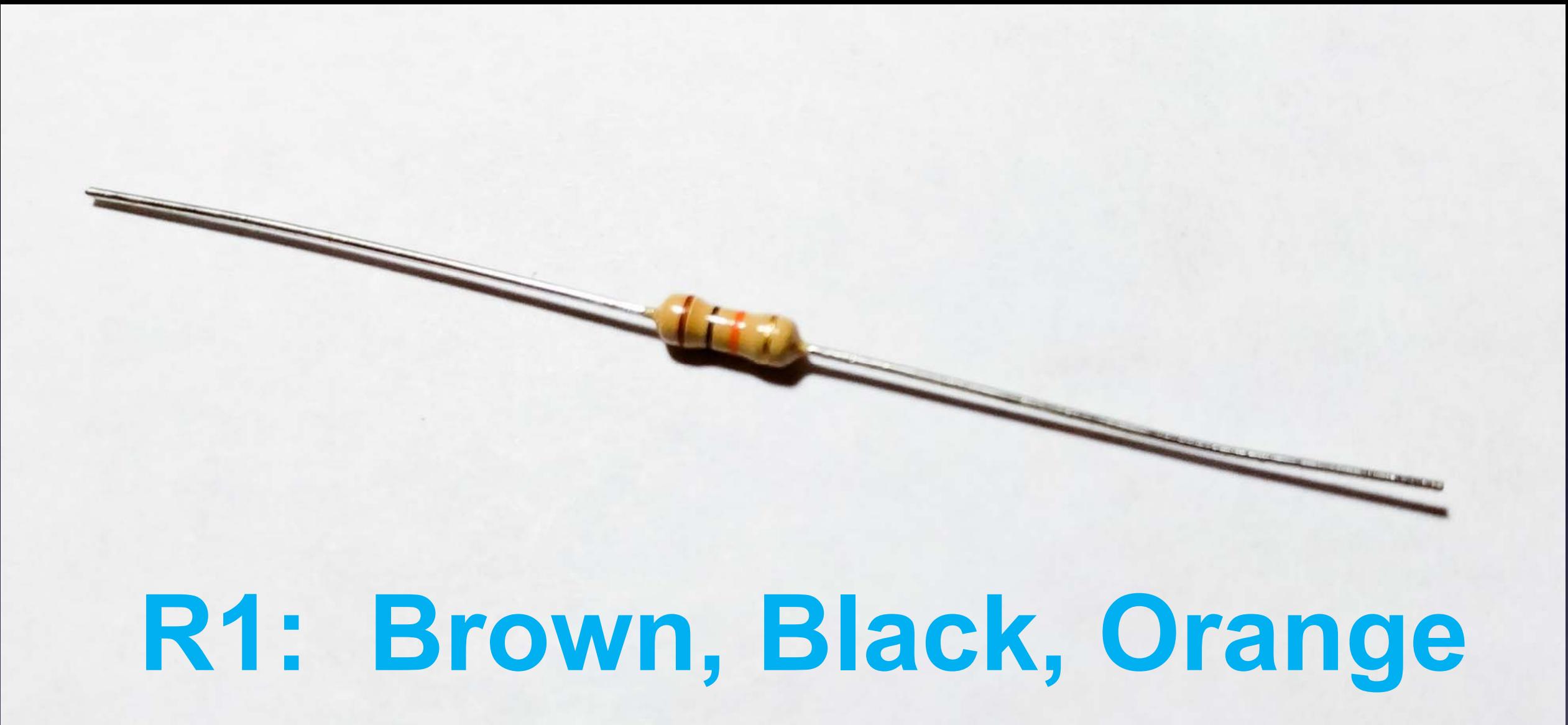


Note:
Most
lead-free solder
has poisonous fumes!

The tools you'll need:

- soldering Iron (35W or less) (0.7mm)
- solder (60/40 Sn/Pb, rosin core, 0.031" diameter or less)
(63/37 is also good)
- soldering iron stand
- cellulose kitchen sponge (*not plastic!*)
- *small* wire cutter

Our first part



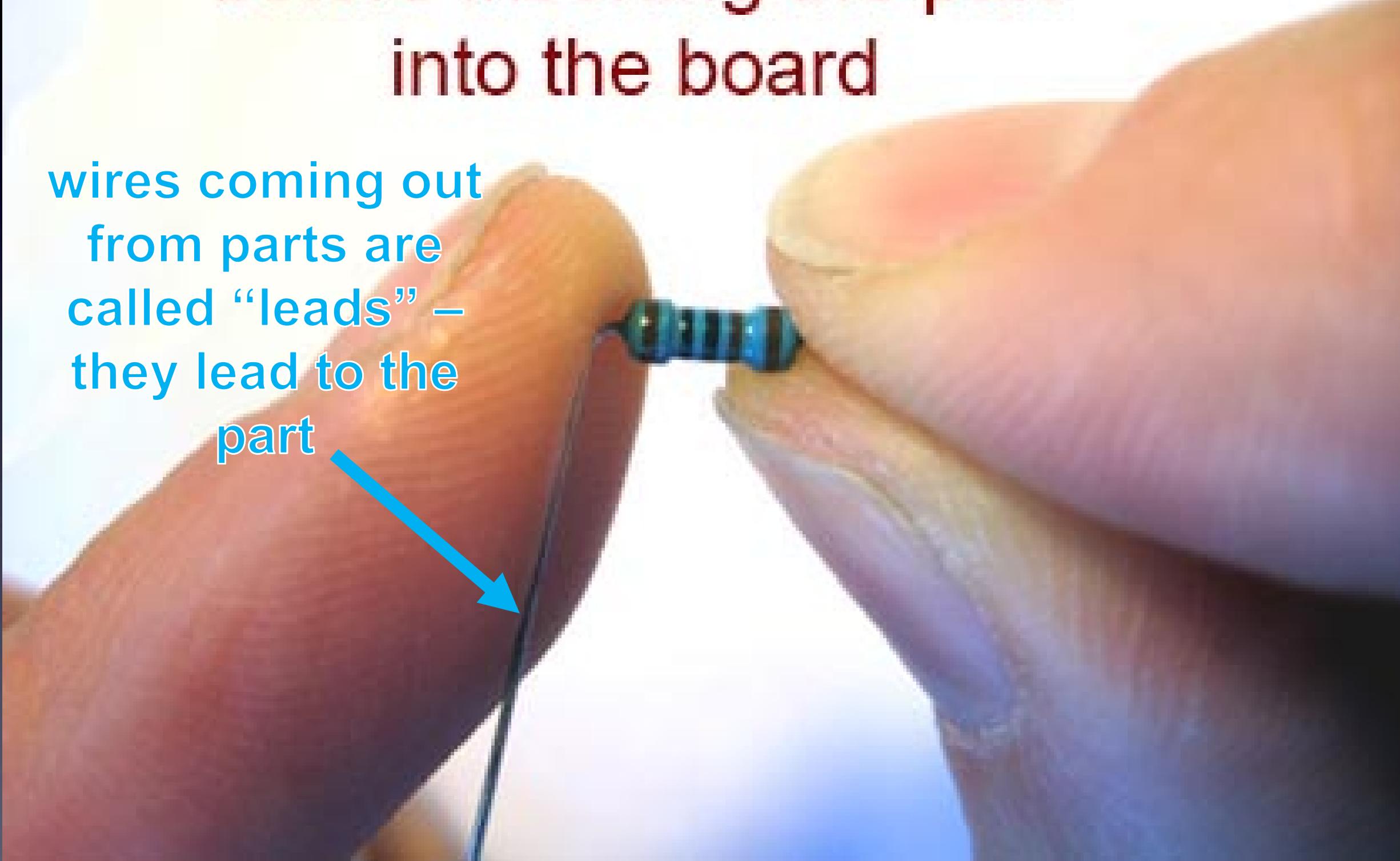
R1: Brown, Black, Orange

(not Brown, Black, Red)

Some parts, such as resistors, need their leads bent first

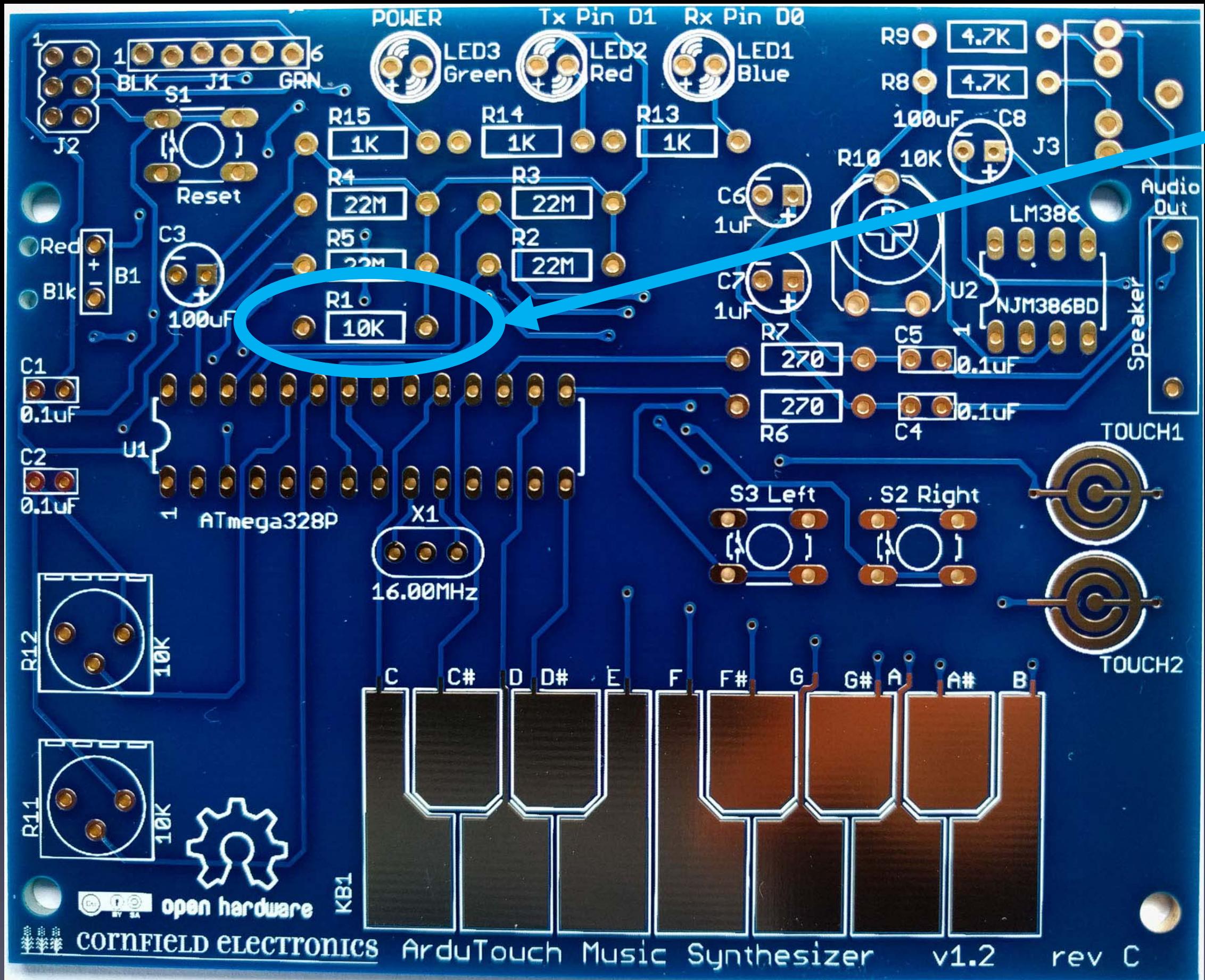
If necessary, Bend leads
before inserting the part
into the board

wires coming out
from parts are
called “leads” –
they lead to the
part





R1 – this is how it will look *before* inserting it into the board



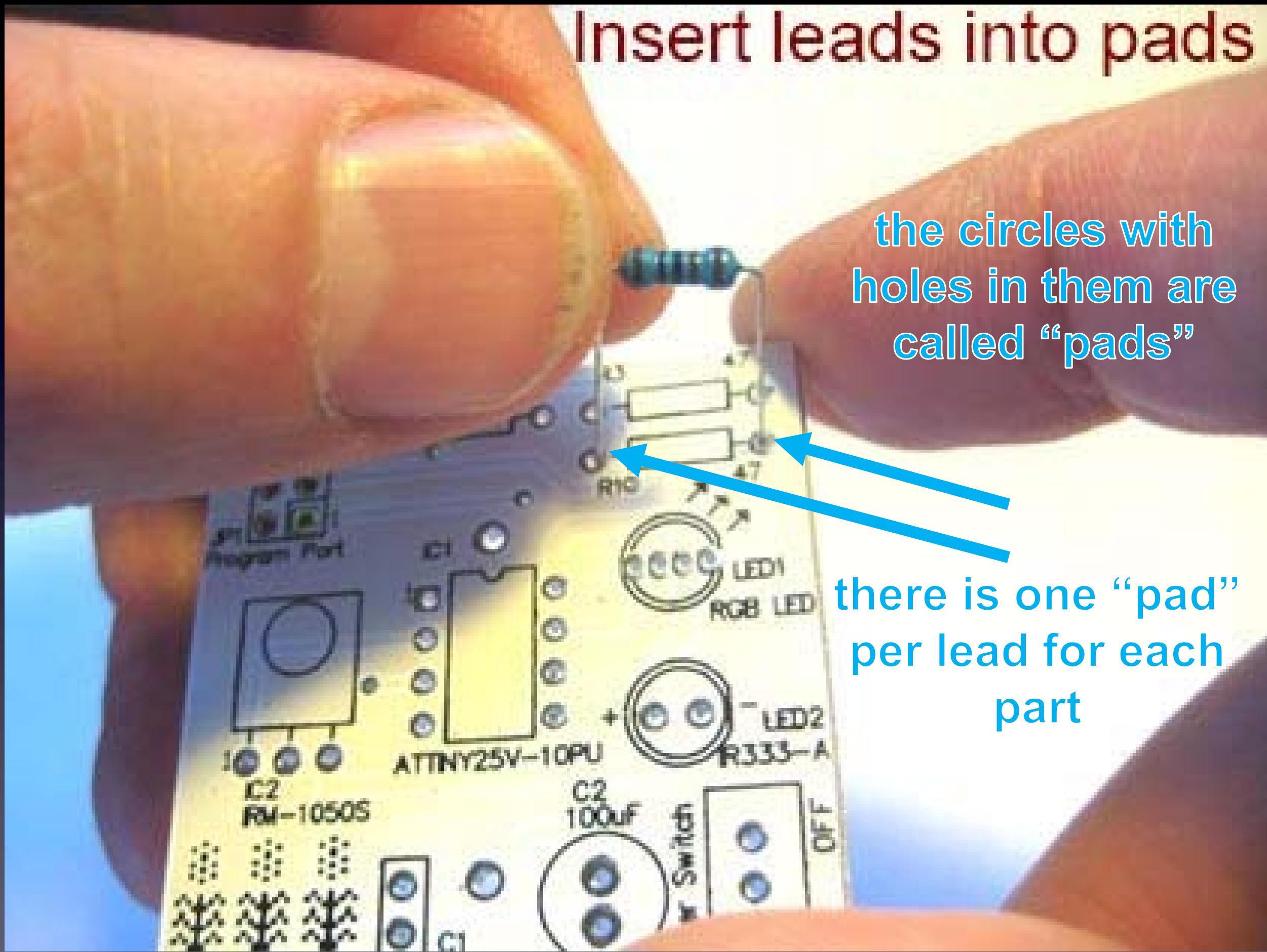
R1

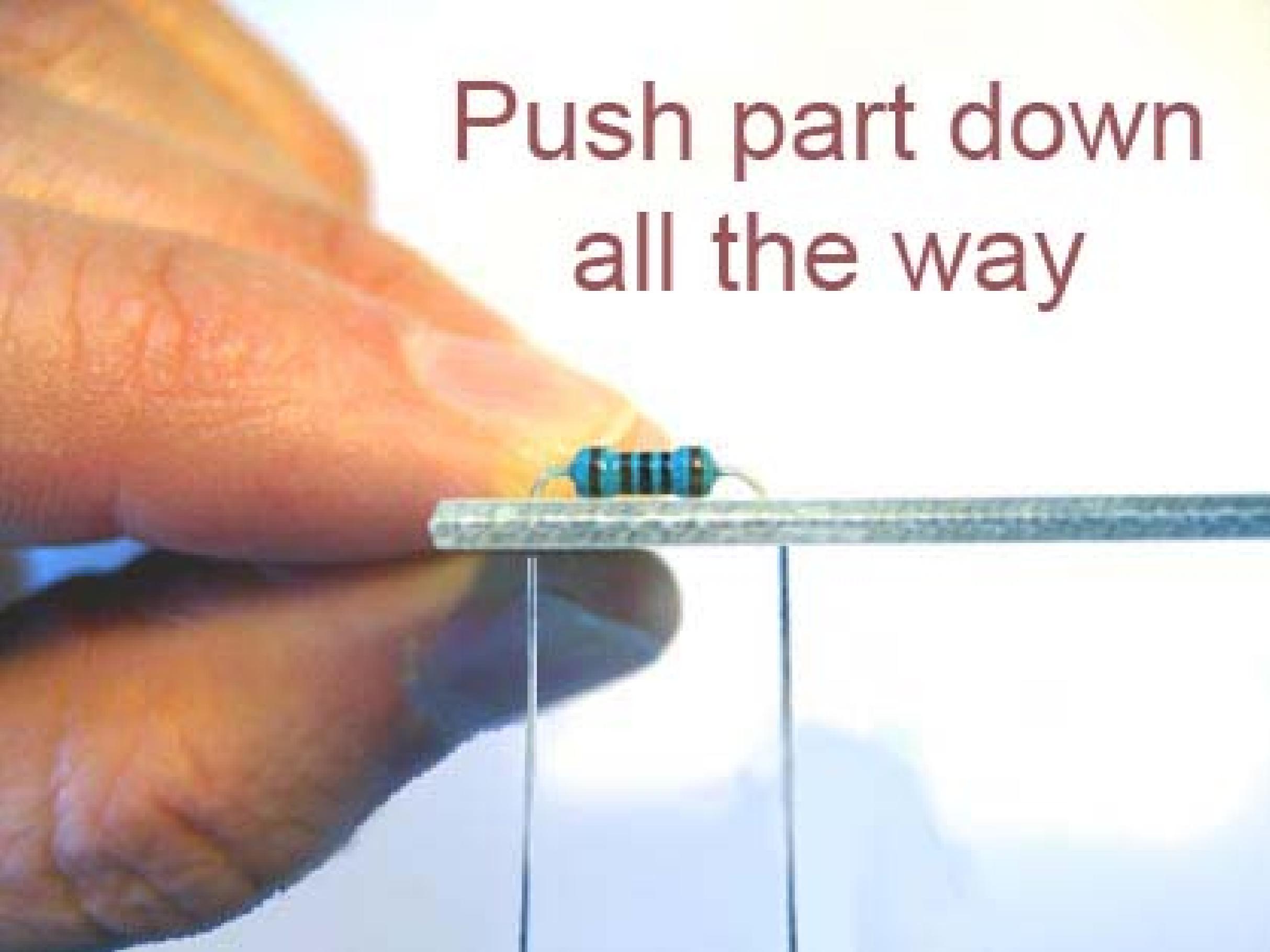
R1 - this is where it goes

Insert leads into pads

the circles with
holes in them are
called “pads”

there is one “pad”
per lead for each
part



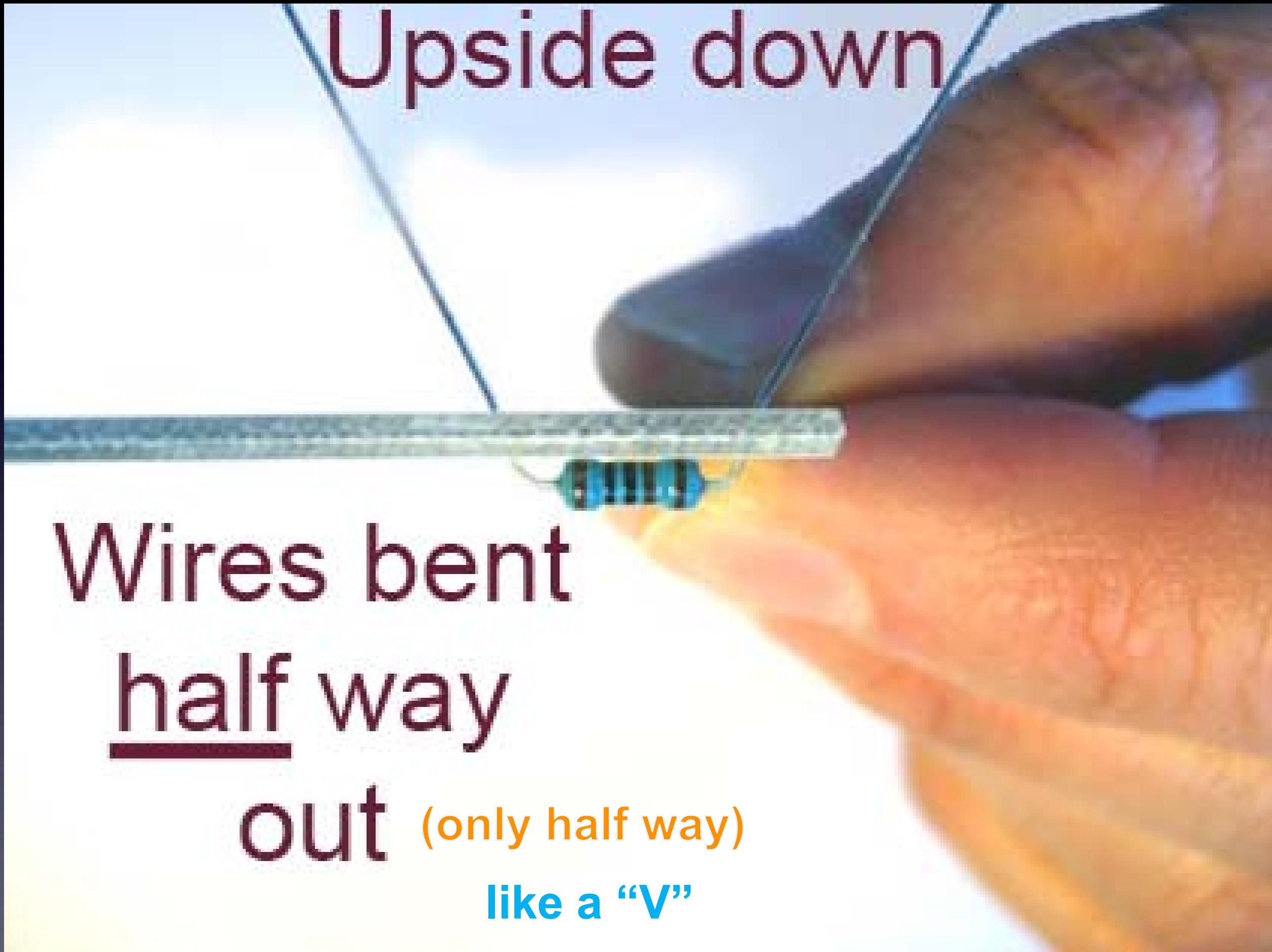


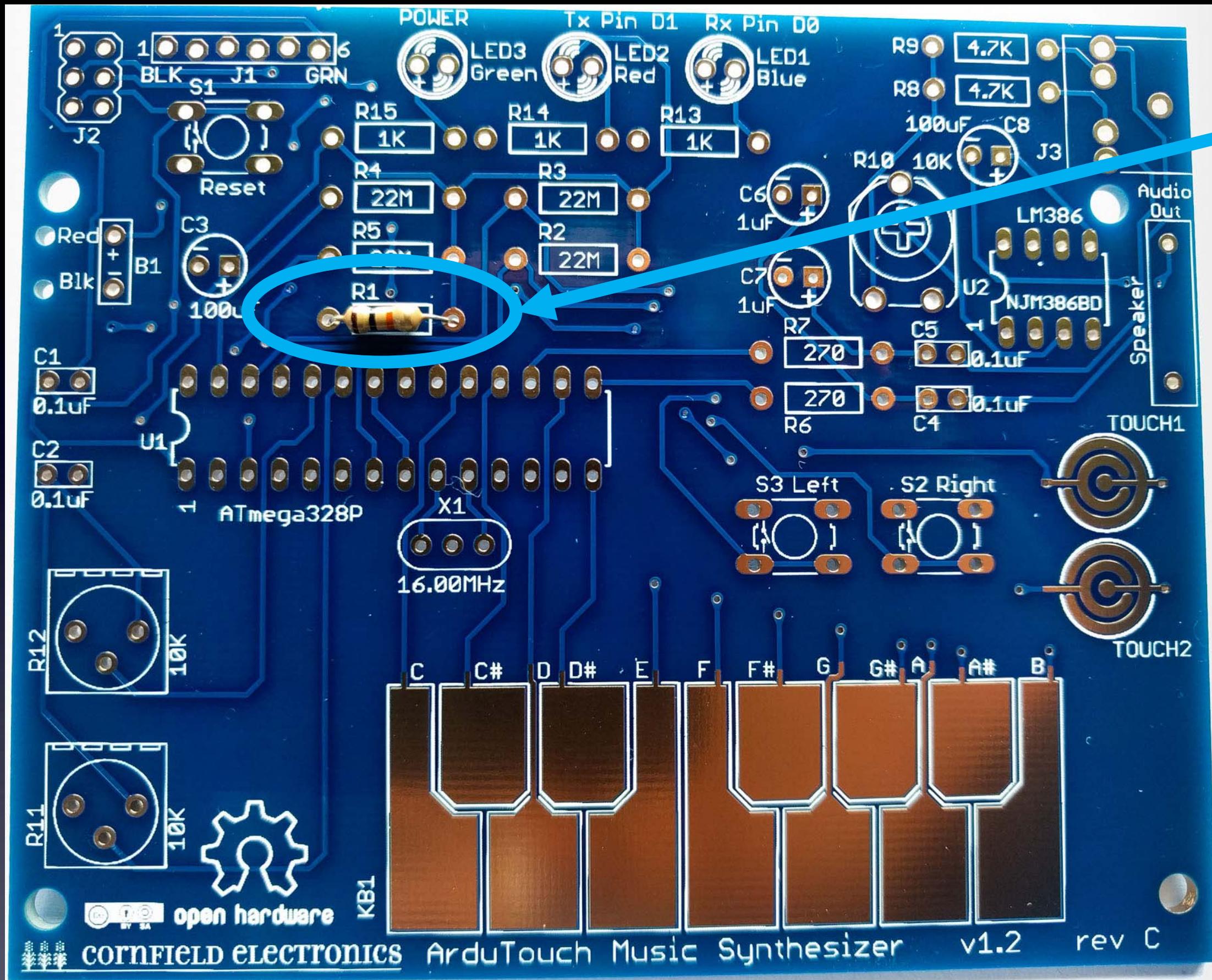
Push part down
all the way

Upside down

Wires bent
half way
out (only half way)
like a “V”

so that the part won't fall out while soldering it





R1 - inserted into the board



How to hold a soldering iron

(Like a pencil – held from underneath)

Important

The perfect kind of solder for
electronics:

60/40 rosin core,
0.031" (0.7mm) diameter (or smaller)

(63/37 is also good)

Note:

Most
lead-free solder
has poisonous fumes!

The perfect kind of solder for electronics:

*This is the only good Lead-Free solder I have found!
(after years of searching)*



**Chip Quik Germanium-Doped Solder
Sn/Cu0.7/Ni0.05/Ge0.006**

3 Safety Tips...

Safety Tip #1:

Hot !!

(When you touch the tip,
you *will* let go quickly every time!)

Safety Tip #2:

Lead (Pb) is toxic

But it easily washes off your hands
with soap and water

Safety Tip #3:

(coming soon)

2 secrets
to good soldering...

Secret #1:

Clean the tip!

(before every solder connection)

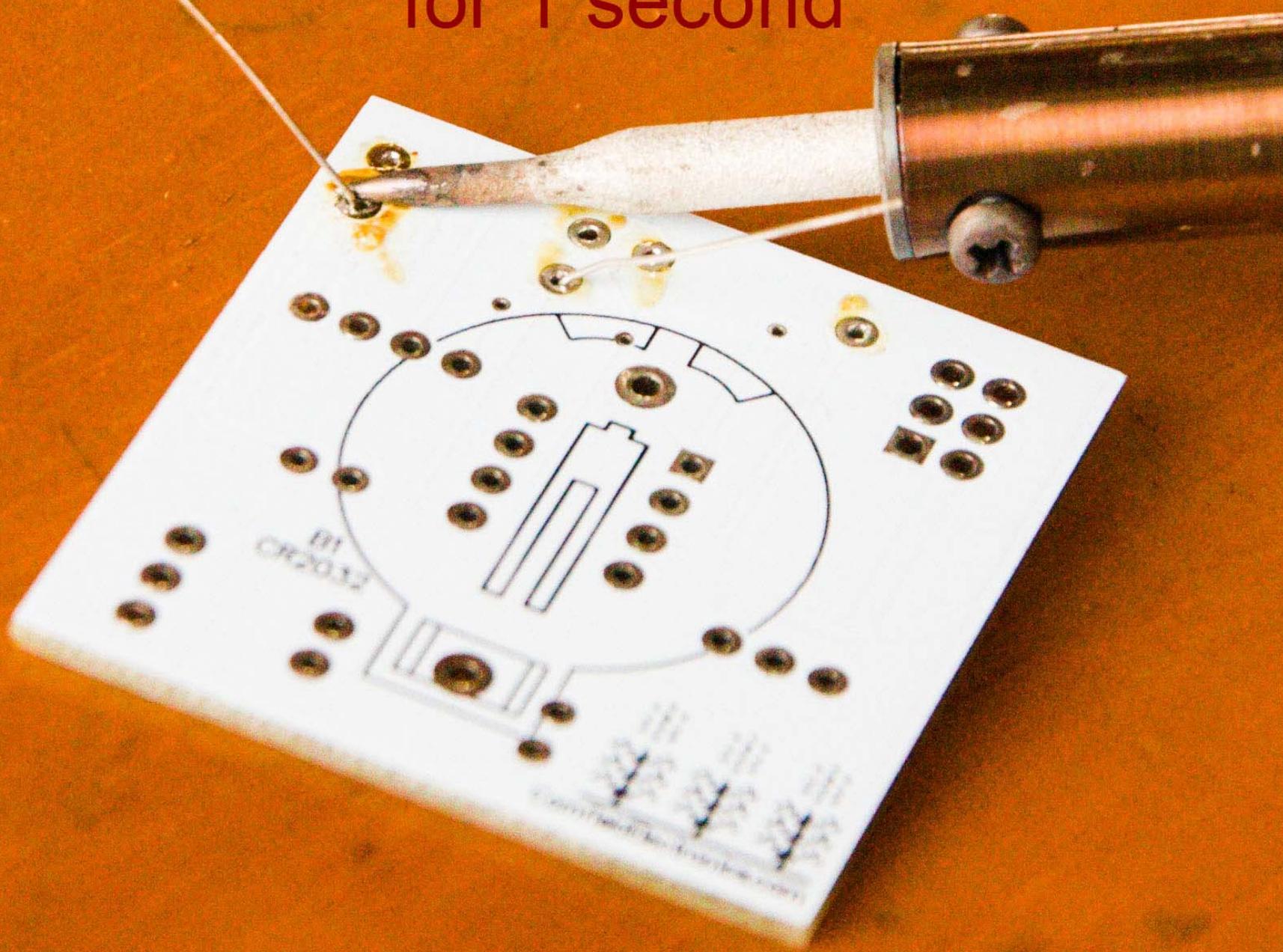
Bang (lightly) 3 times,

Swipe, Rotate, Swipe (on the sponge):

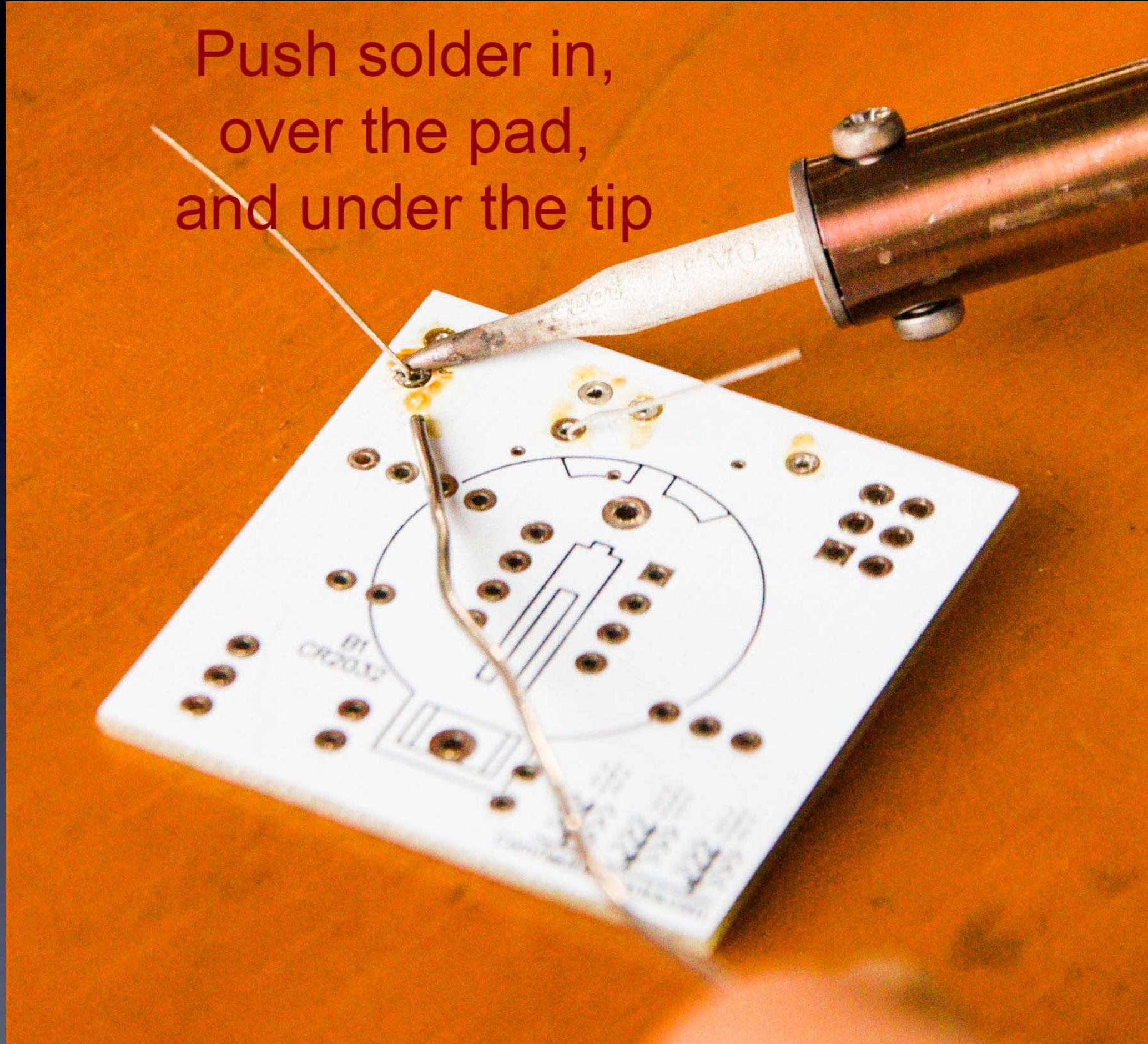
Keep the tip shiny silver!

knock solder off the tip

Lay clean tip across half of the pad,
touching the pad and lead
for 1 second

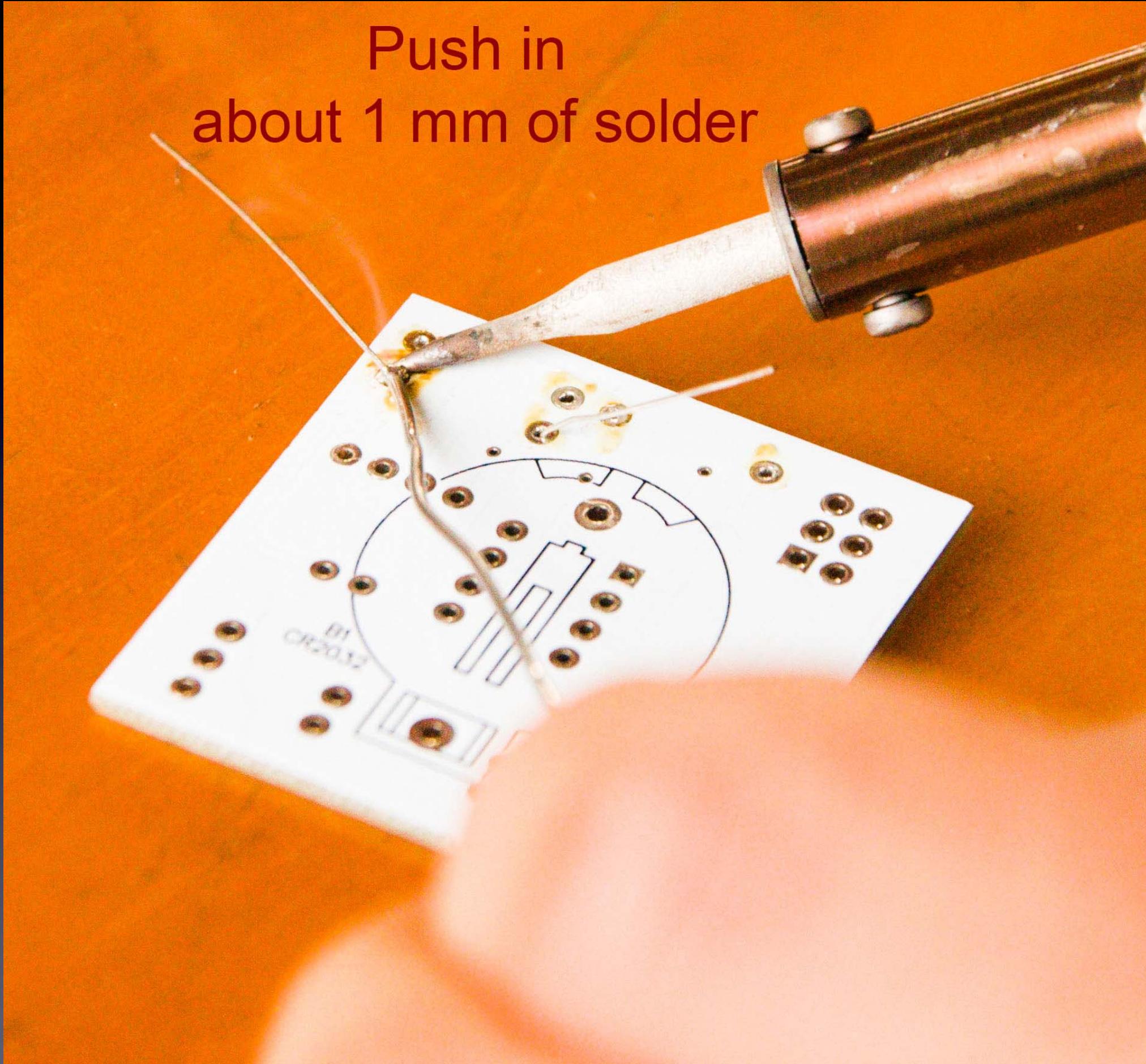


Do this quickly (slowly doesn't work well) – solder in & out in about 1 second



Make sure solder melts on the underside of the soldering iron tip
(not the side or top of the soldering iron tip)!

Do this quickly (slowly doesn't work well) – solder in & out in about 1 second



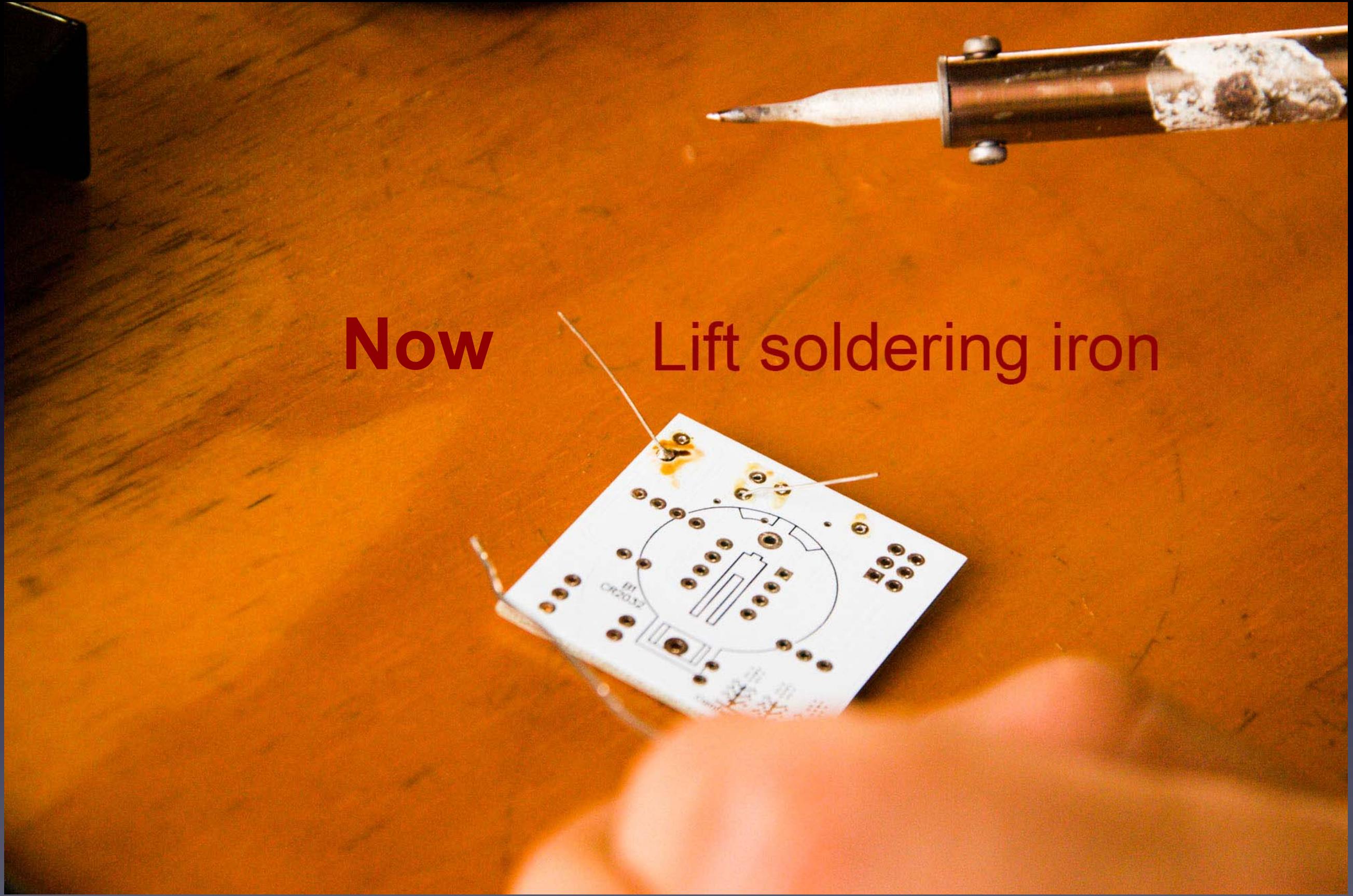
Make sure solder melts on the underside of the soldering iron tip
(not the side or top of the soldering iron tip)!



Pull solder away,
But keep holding soldering iron down
for 1 more second

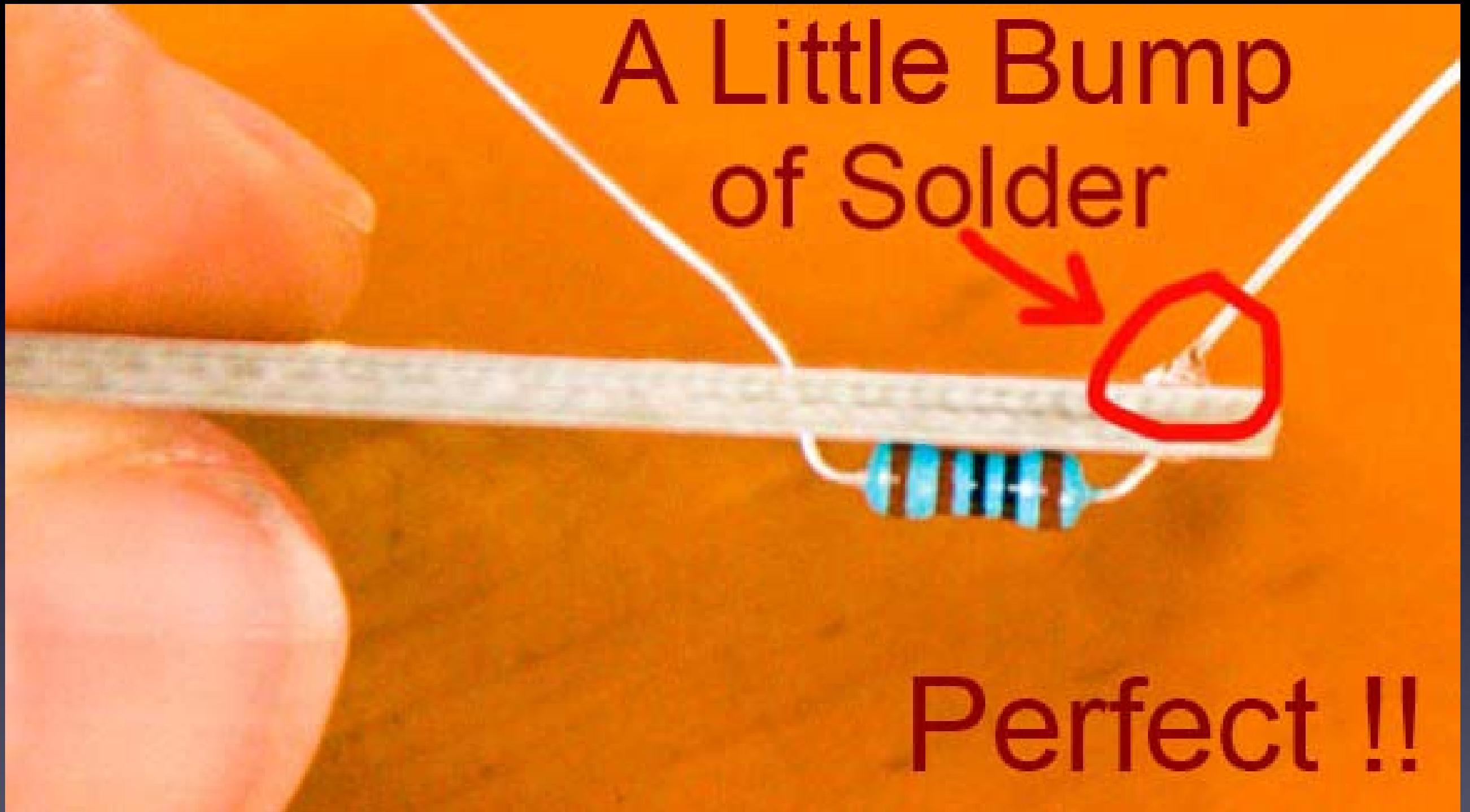
Secret #2:

Keep hot tip down
1 second
for solder to flow !!



Now

Lift soldering iron



A Little Bump
of Solder

Perfect !!

If you can see any of the pad, or the hole, you need more solder – so, just do all the steps again to make it perfect.

The Rhythm !
and speed (about 1 second per step)



The Rhythm !
and speed (about 1 second per step)
Clean the tip



The Rhythm !
and speed (about 1 second per step)



Tip Down

The Rhythm !
and speed (about 1 second per step)



Solder In

The Rhythm !
and speed (about 1 second per step)



Solder Out

The Rhythm !
and speed (about 1 second per step)



WAIT !

The Rhythm !
and speed (about 1 second per step)



Lift Tip

The Rhythm !
and speed (about 1 second per step)



The Rhythm !
and speed (about 1 second per step)
Clean the tip



The Rhythm !
and speed (about 1 second per step)



Tip Down

The Rhythm !
and speed (about 1 second per step)



Solder In

The Rhythm !
and speed (about 1 second per step)



Solder Out

The Rhythm !
and speed (about 1 second per step)



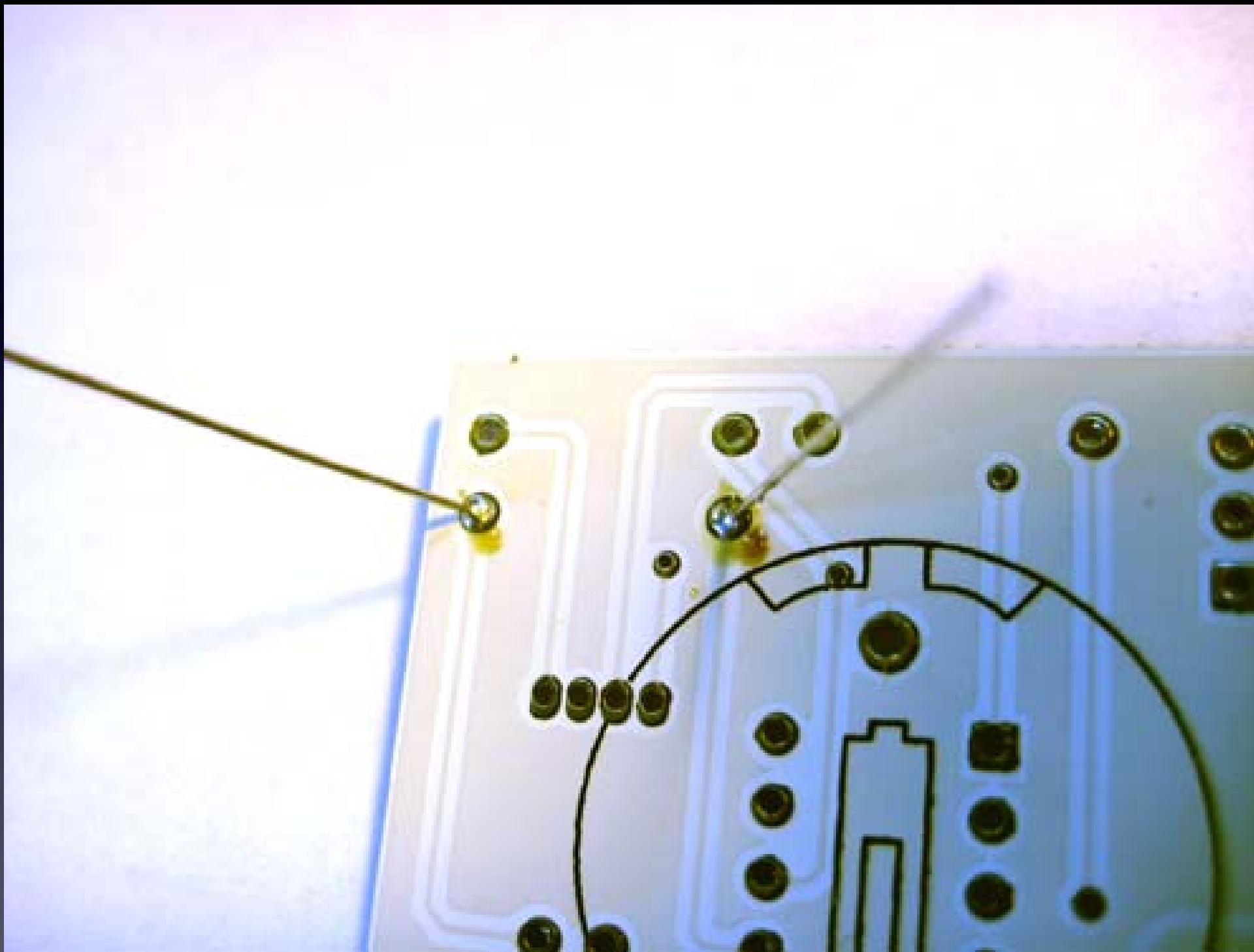
WAIT !

The Rhythm !
and speed (about 1 second per step)



Lift Tip

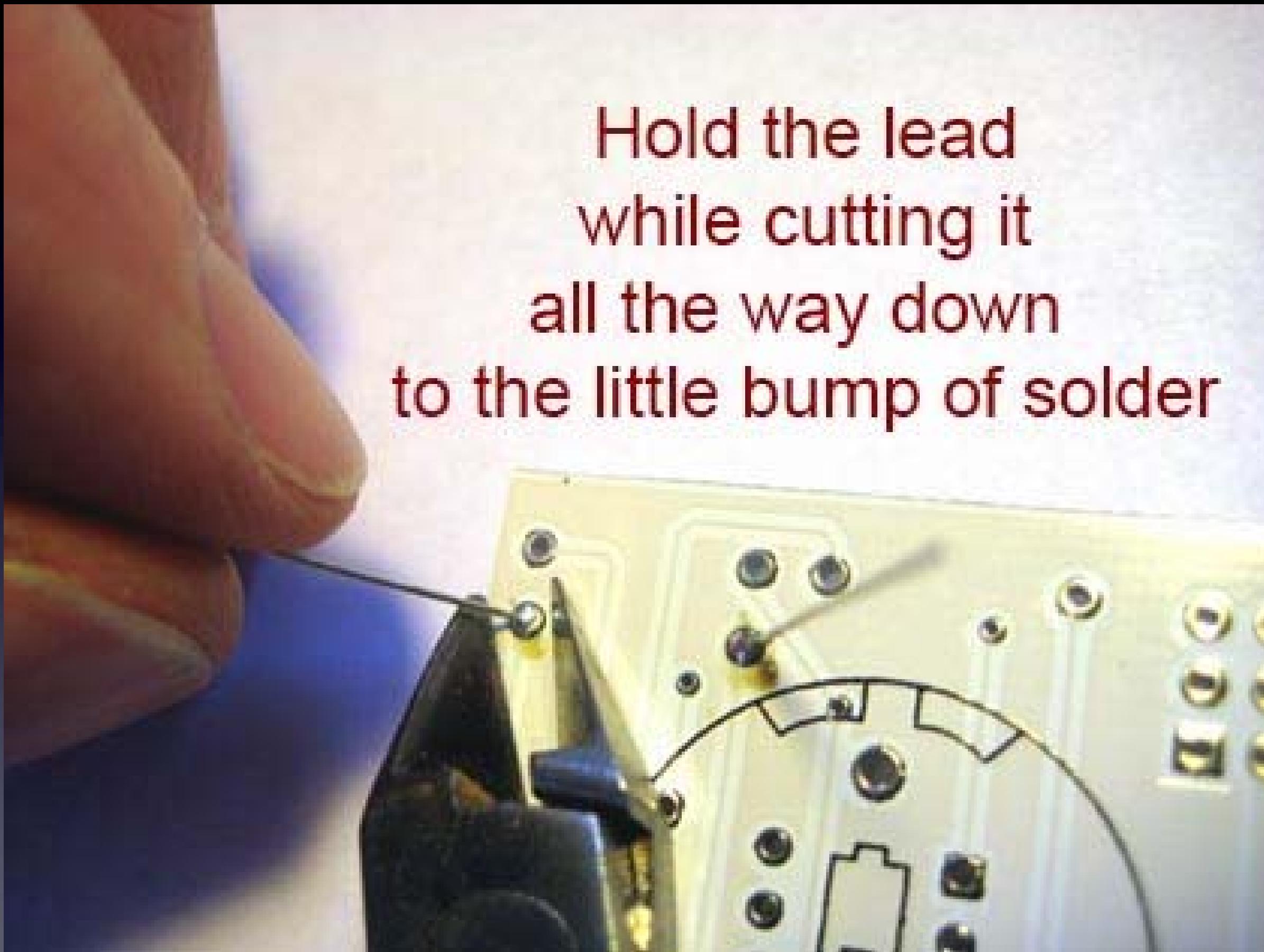
Solder all of the leads of the part to the board



For this part, there are two leads
Here you can see two good solder connections

Now cut the leads short

Hold the lead
while cutting it
all the way down
to the little bump of solder



Cutting with the tip of the wire cutter gives you more control

Safety Tip #3:

Hold or cover the lead !

(or it will fly into your eye!)

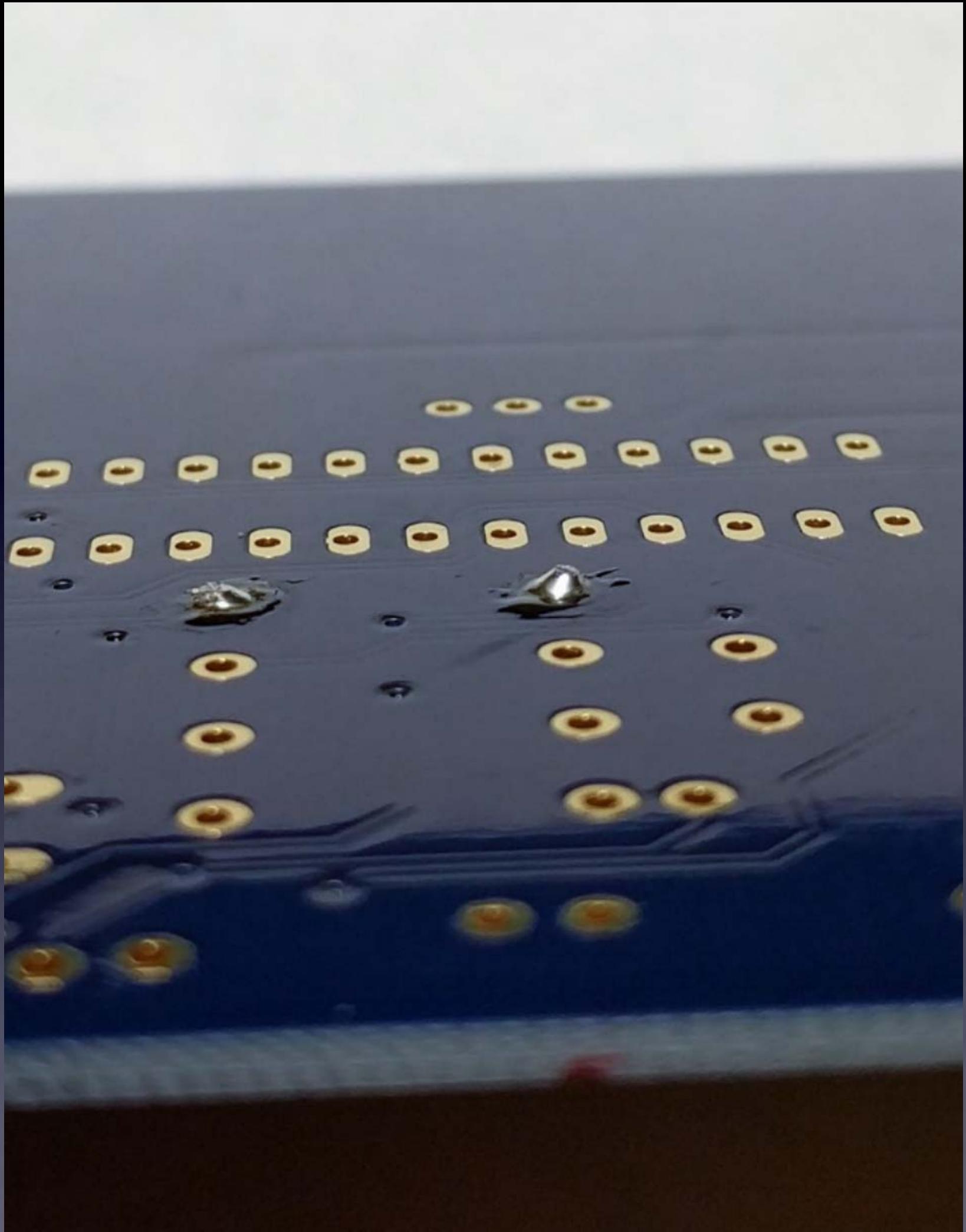
(They like doing that – so please hold or cover the lead when you cut.)



All done !

No wires sticking out

R1 soldered to the board

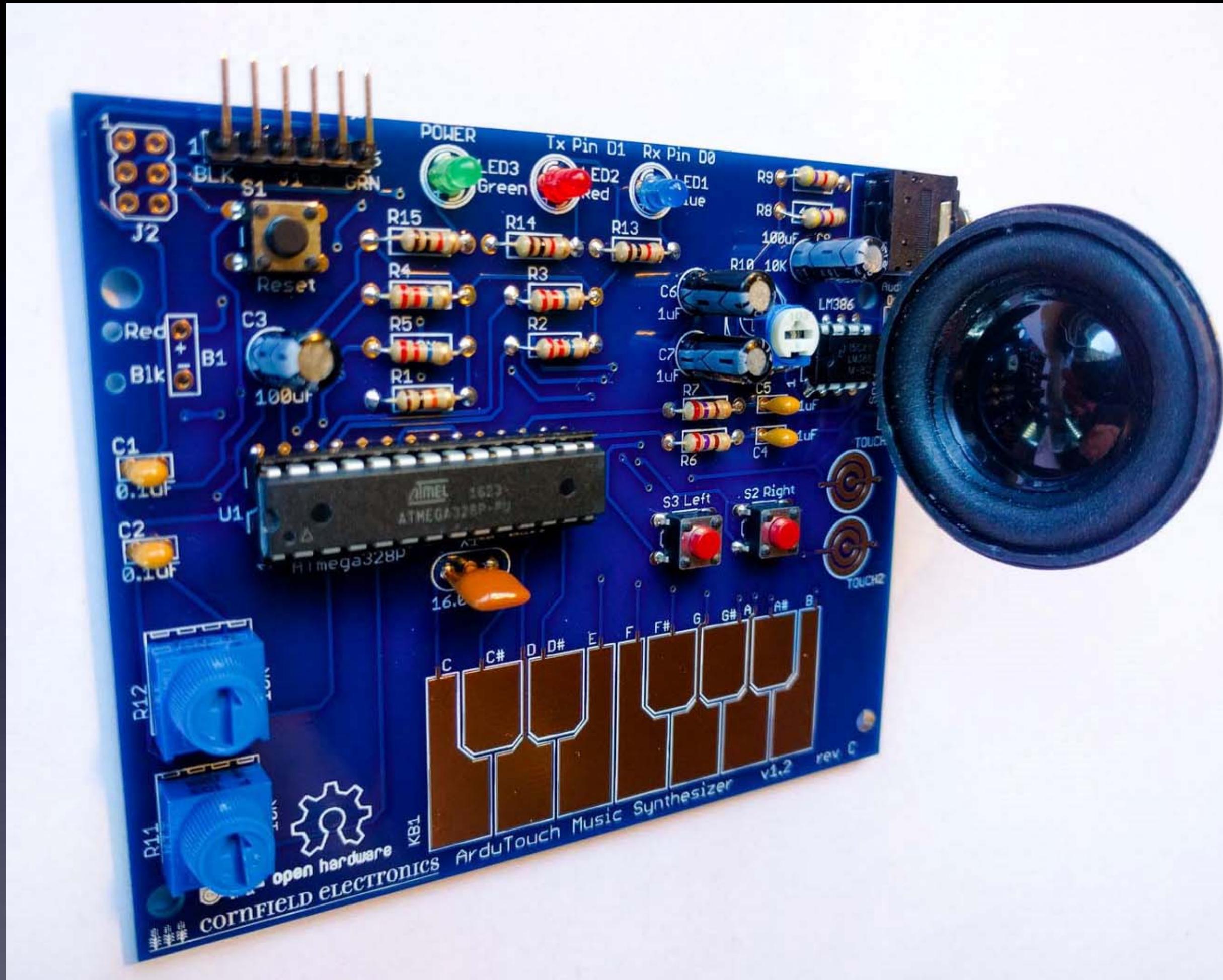


Notice that:

- each connection is a small bump (not flat)
- you cannot see any pad (it's totally covered with solder)
- you cannot see the hole (it's totally covered with solder)

One part at a time

Till all the parts are soldered



And it will look like this when you're done.

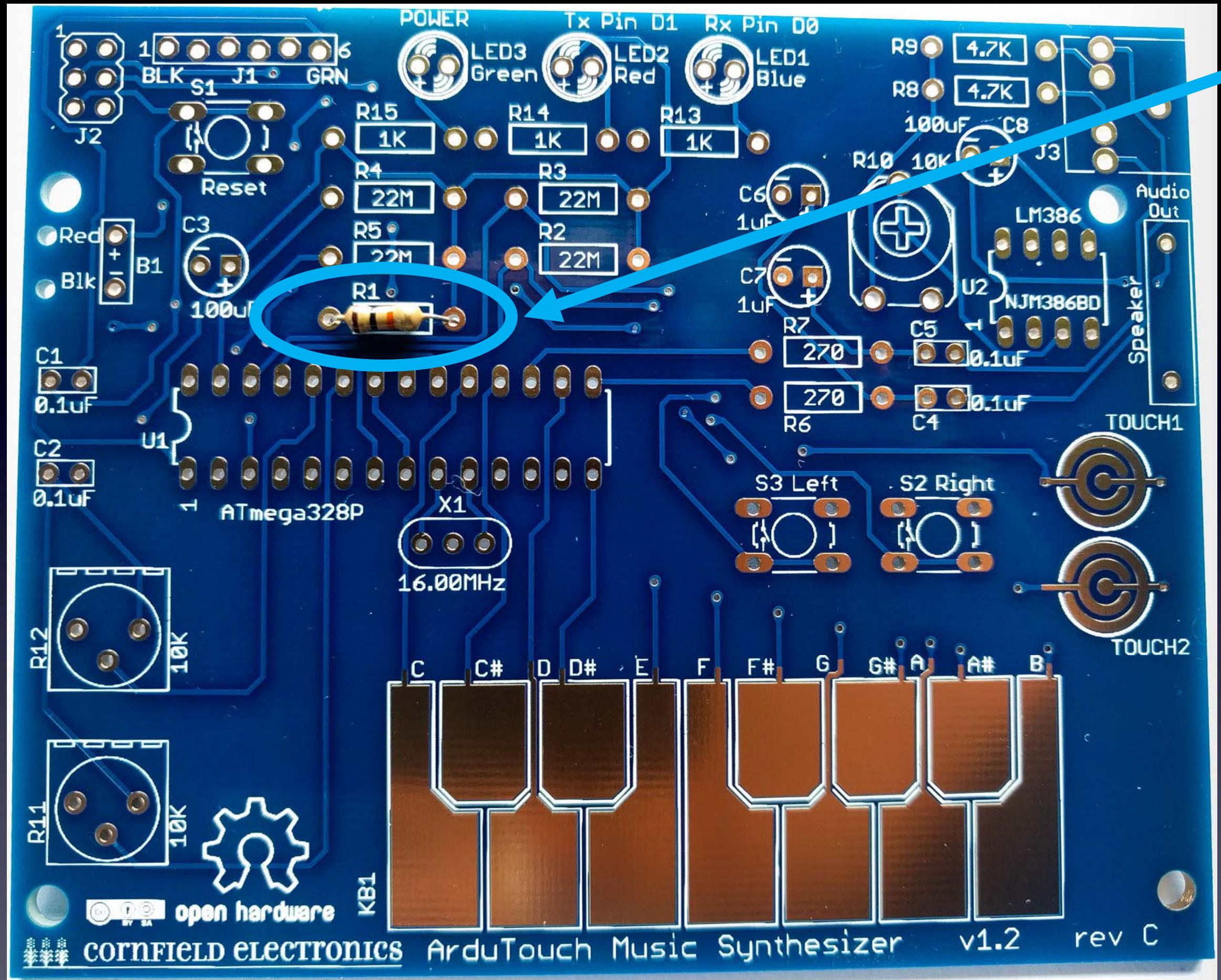
Then put in the batteries,

Turn it on,

And it works!

(Or you start debugging.)

Let's start!



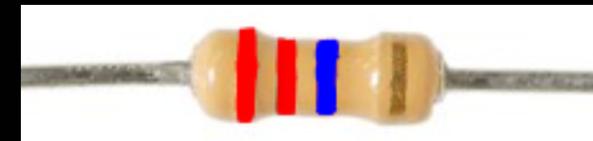
If you haven't done so already, solder R1: brown, black, orange

R1:



10K: Brown, Black, Orange

R2, R3, R4, R5:



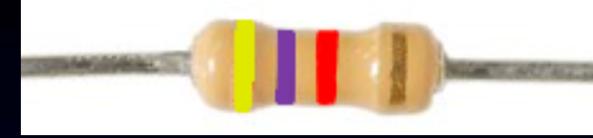
22M: Red, Red, Blue

R6, R7:



270: Red, Violet, Brown

R8, R9:

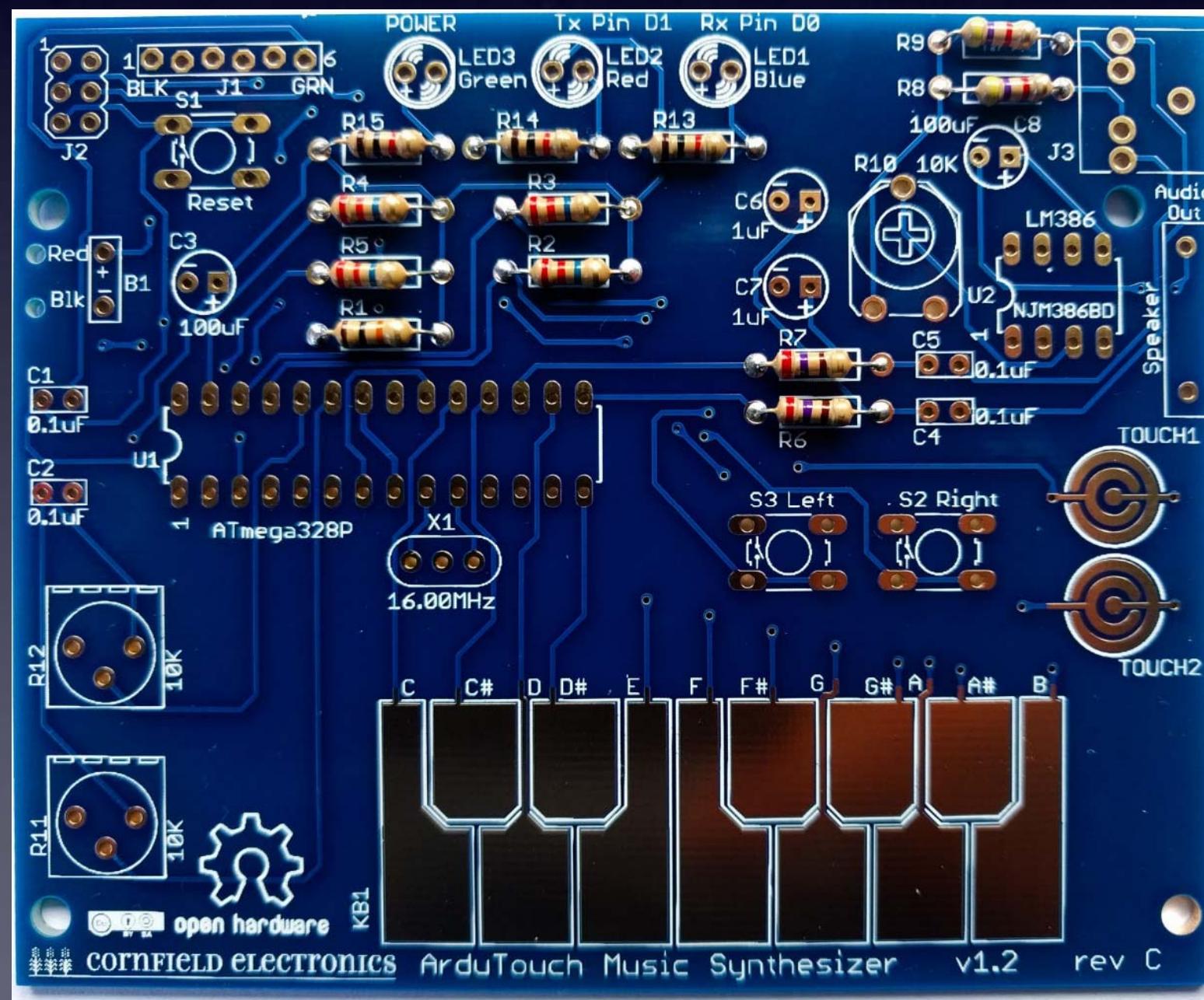


4.7K: Yellow, Violet, Red

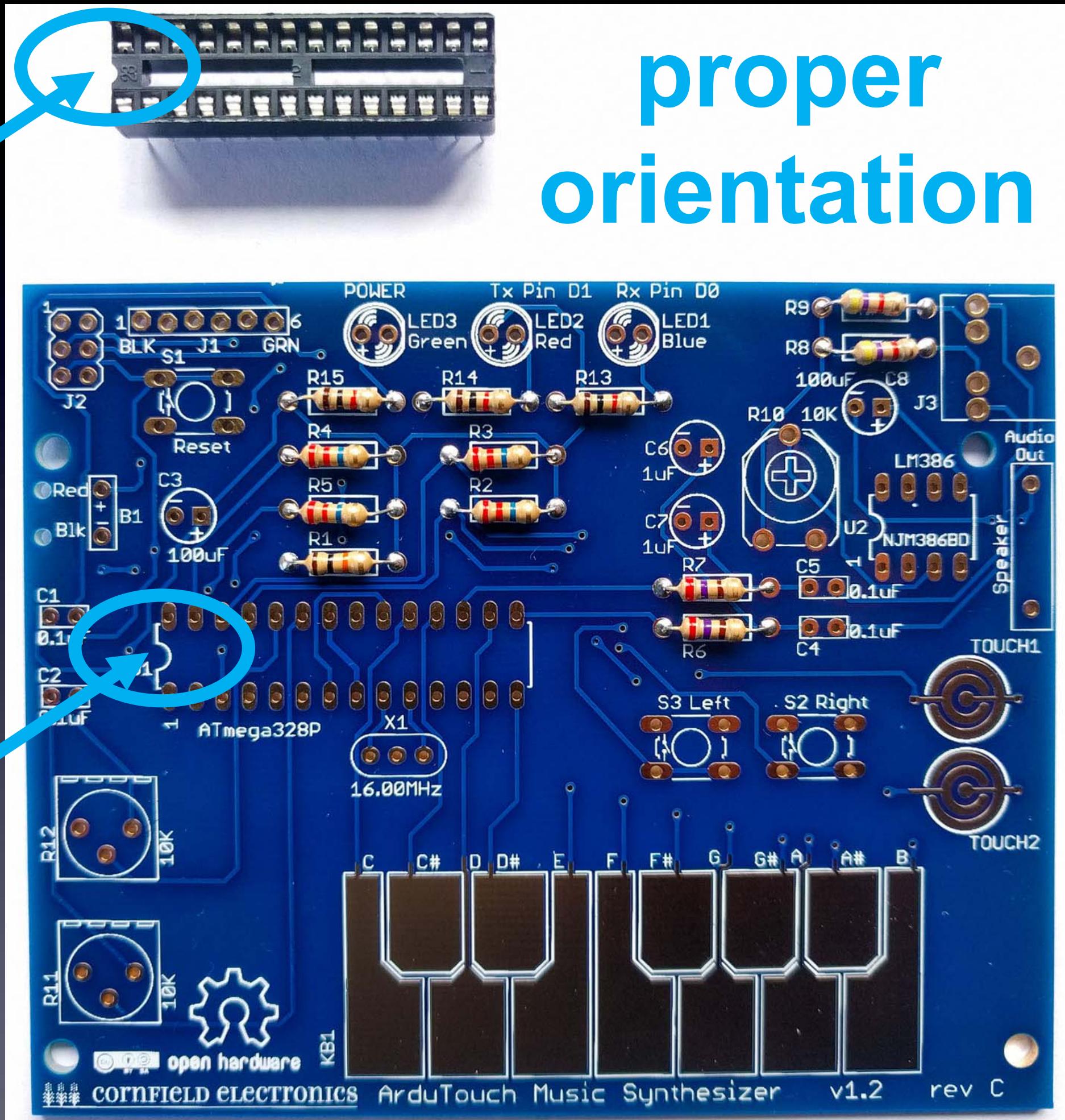
R13, R14, R15:



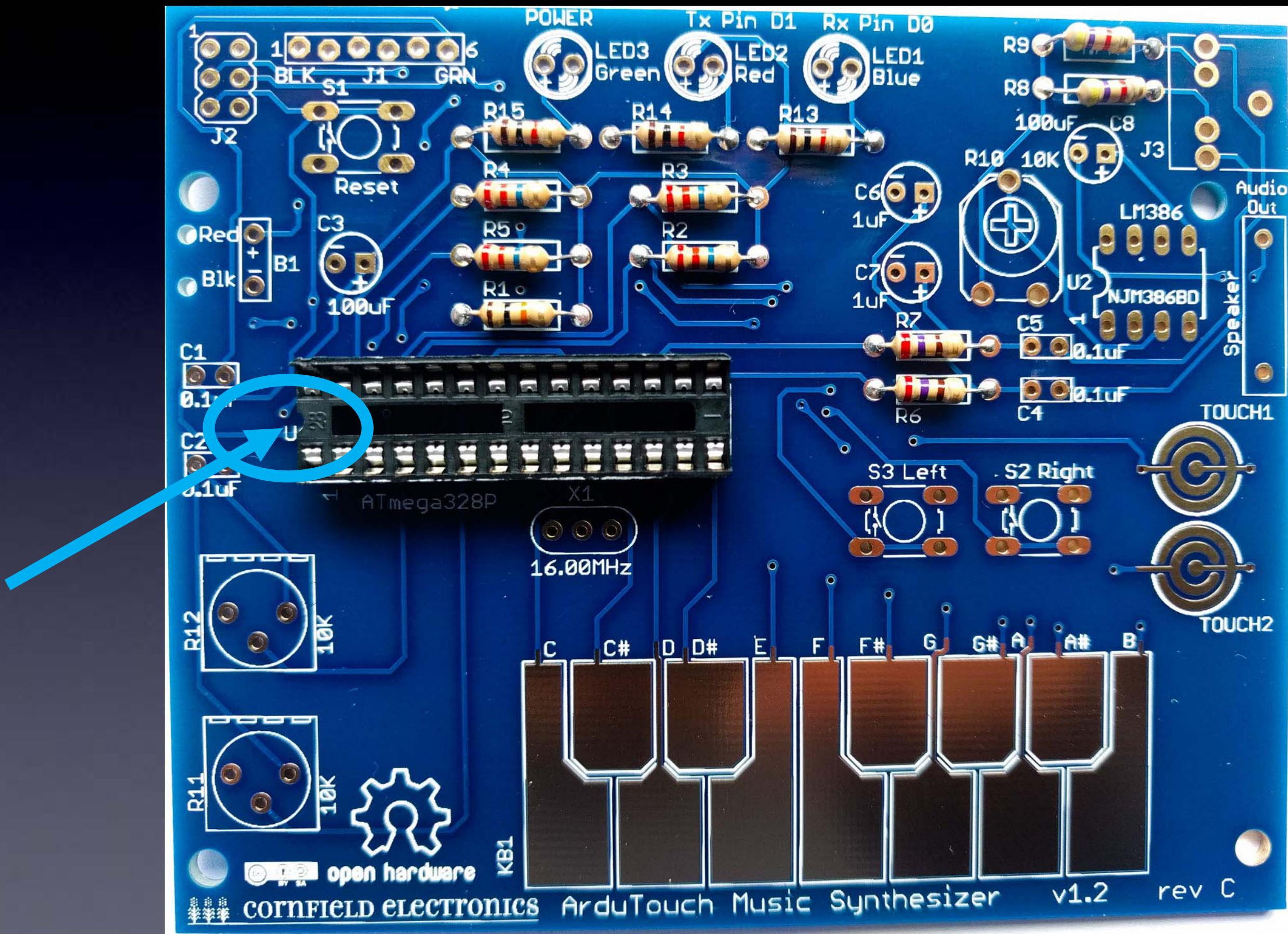
1K: Brown, Black, Red



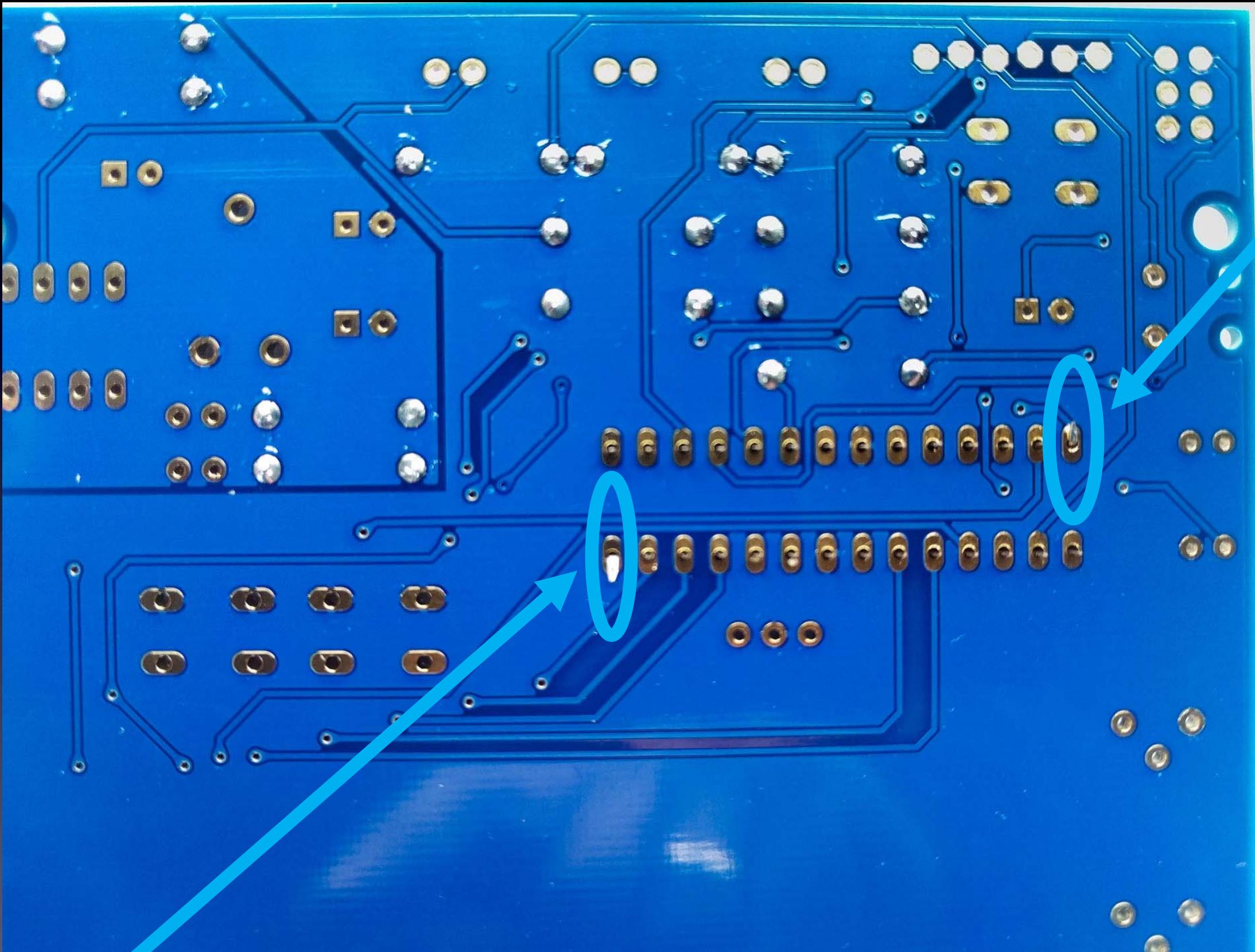
U1: microcontroller socket



U1: microcontroller socket: inserted correctly

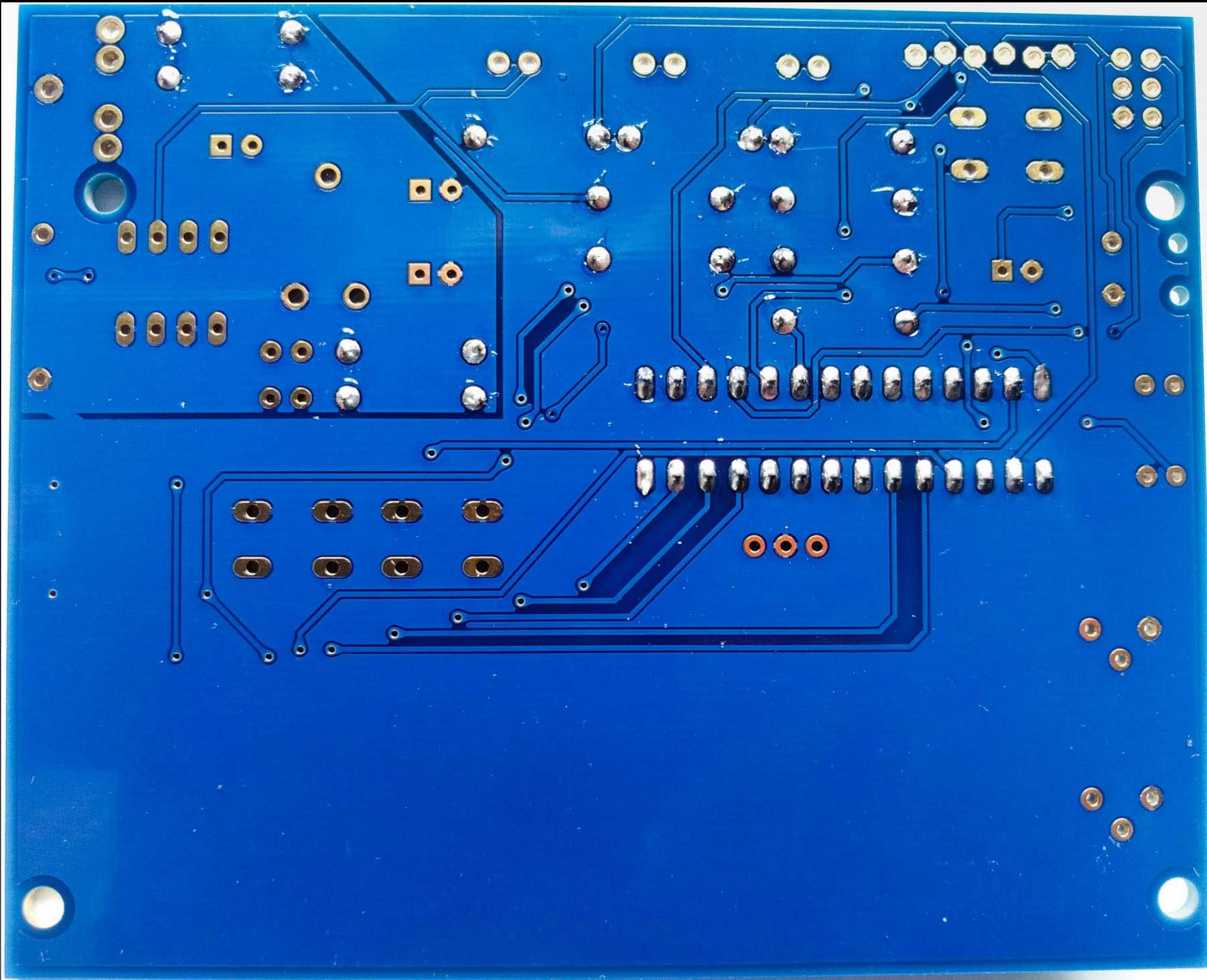


U1: microcontroller socket

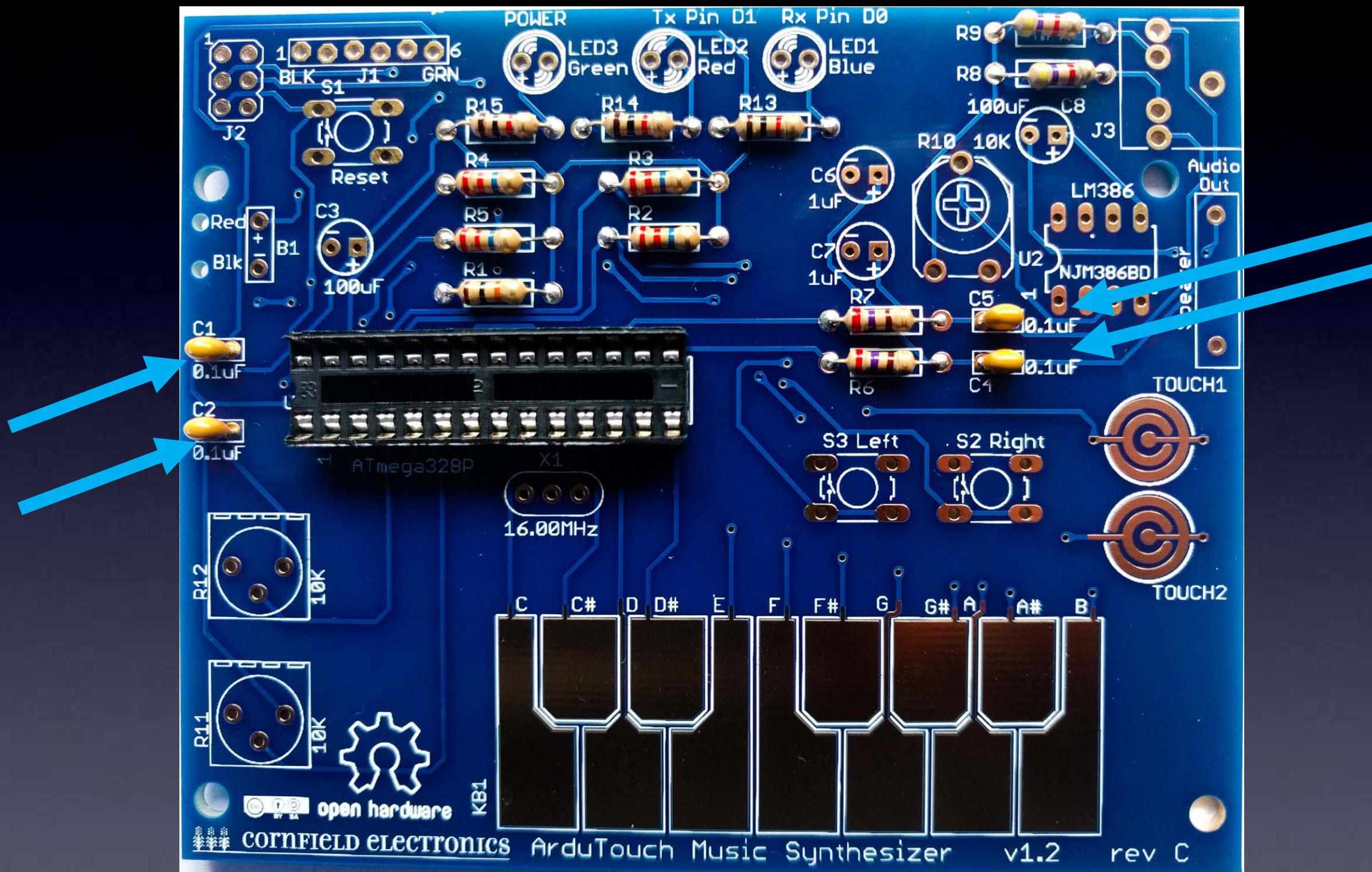


bend pins down on two corners,
and solder all 28 leads to the board

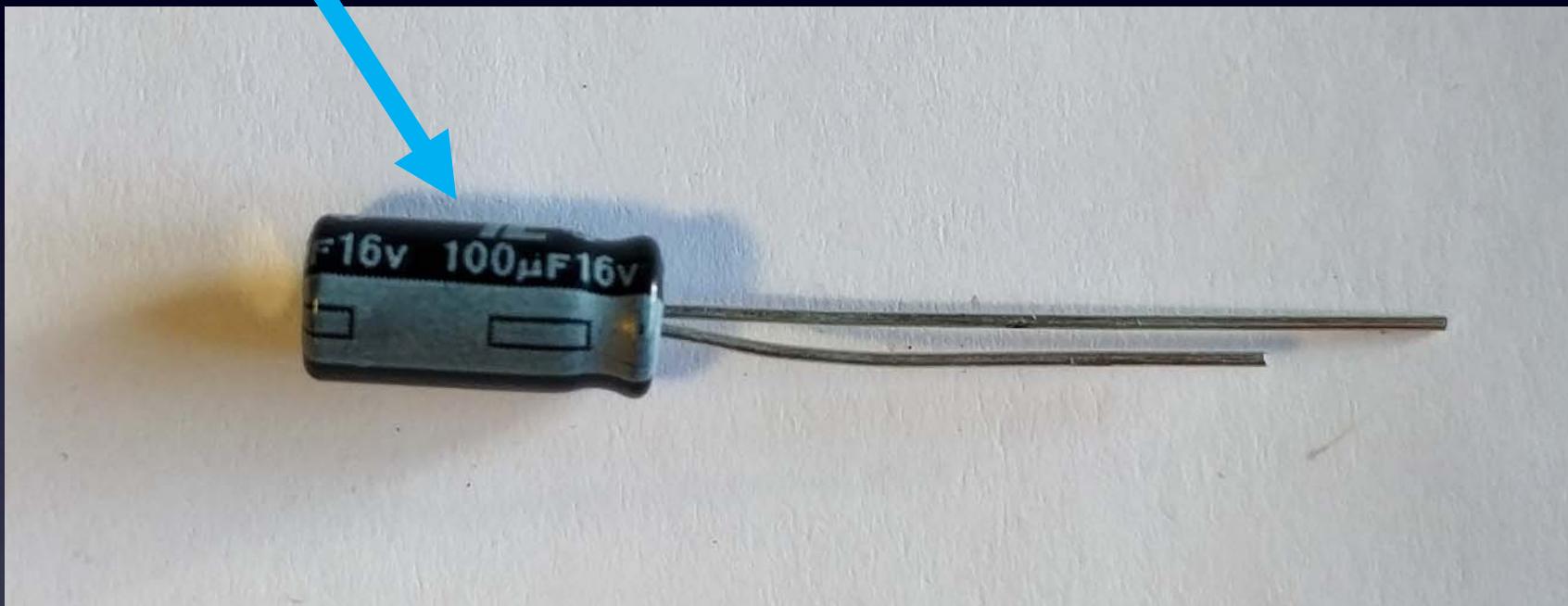
U1: microcontroller socket



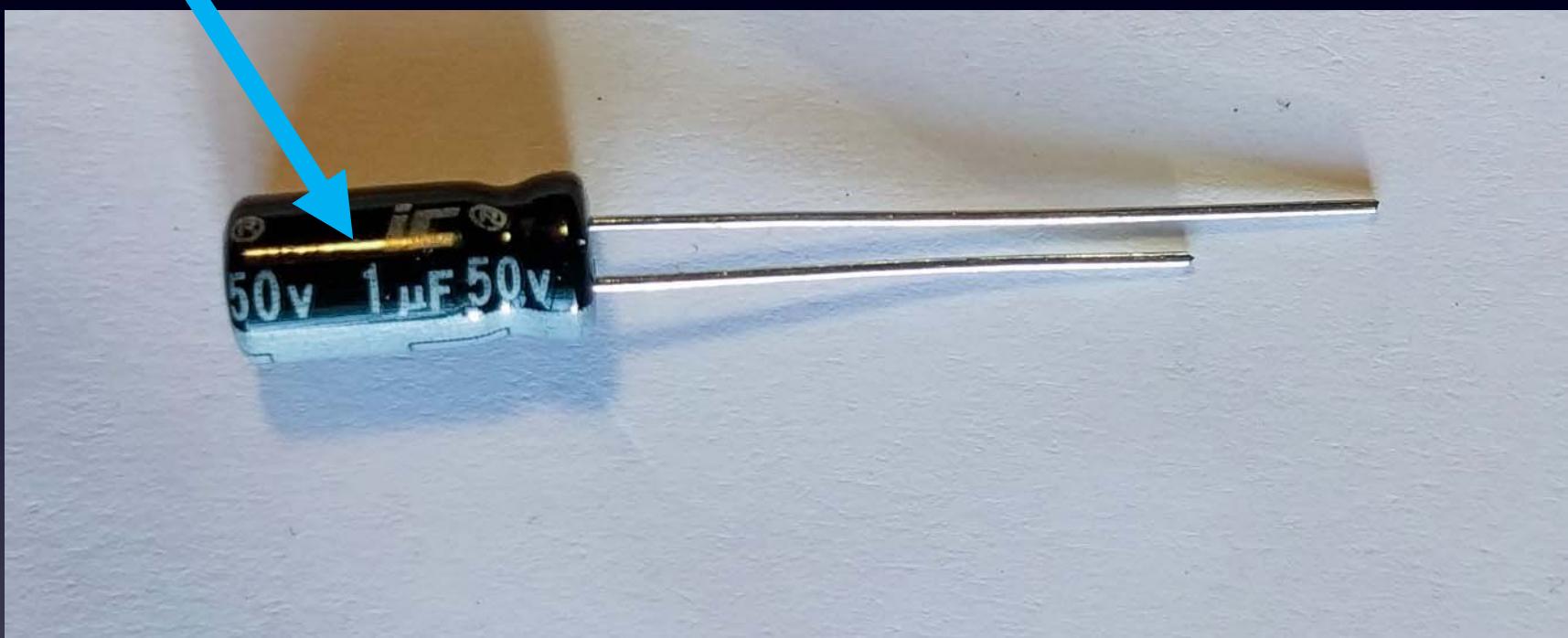
All 28 leads soldered to the board:
→ Notice that each has a little bump of solder (not flat). ←



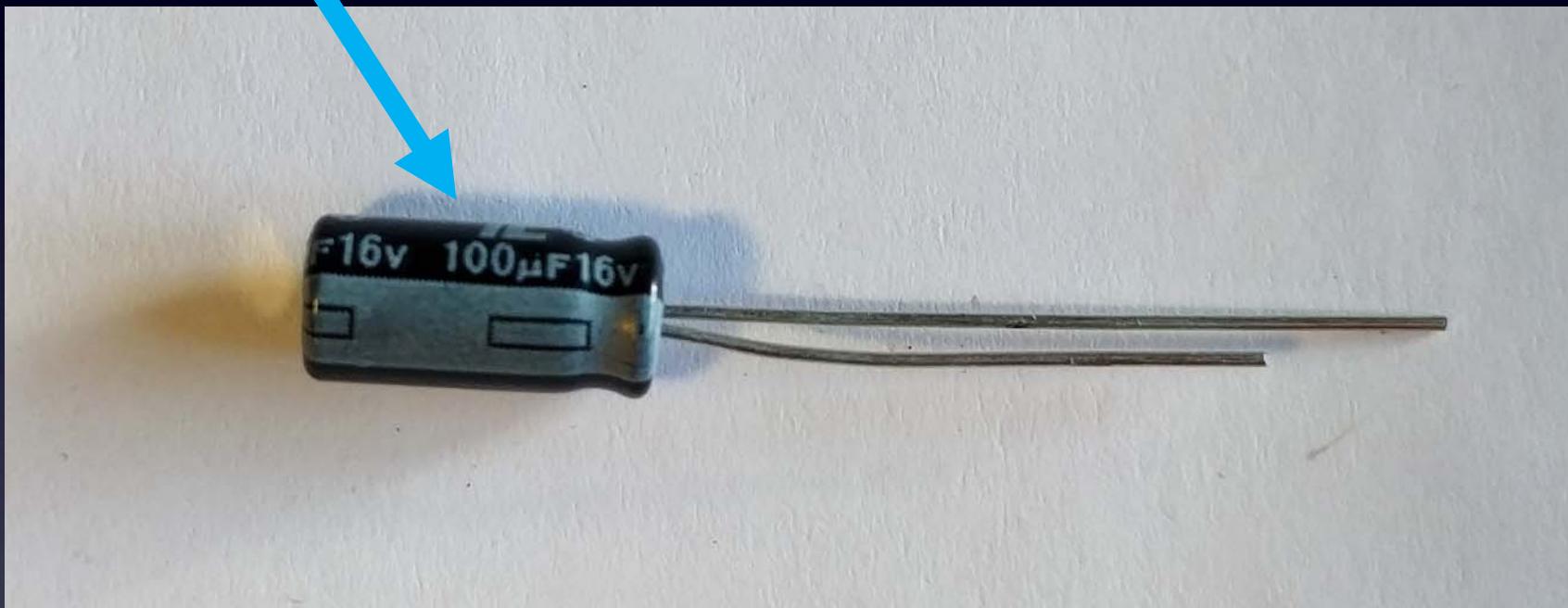
C1, C2, C4, C5



C3, C8: 100uF



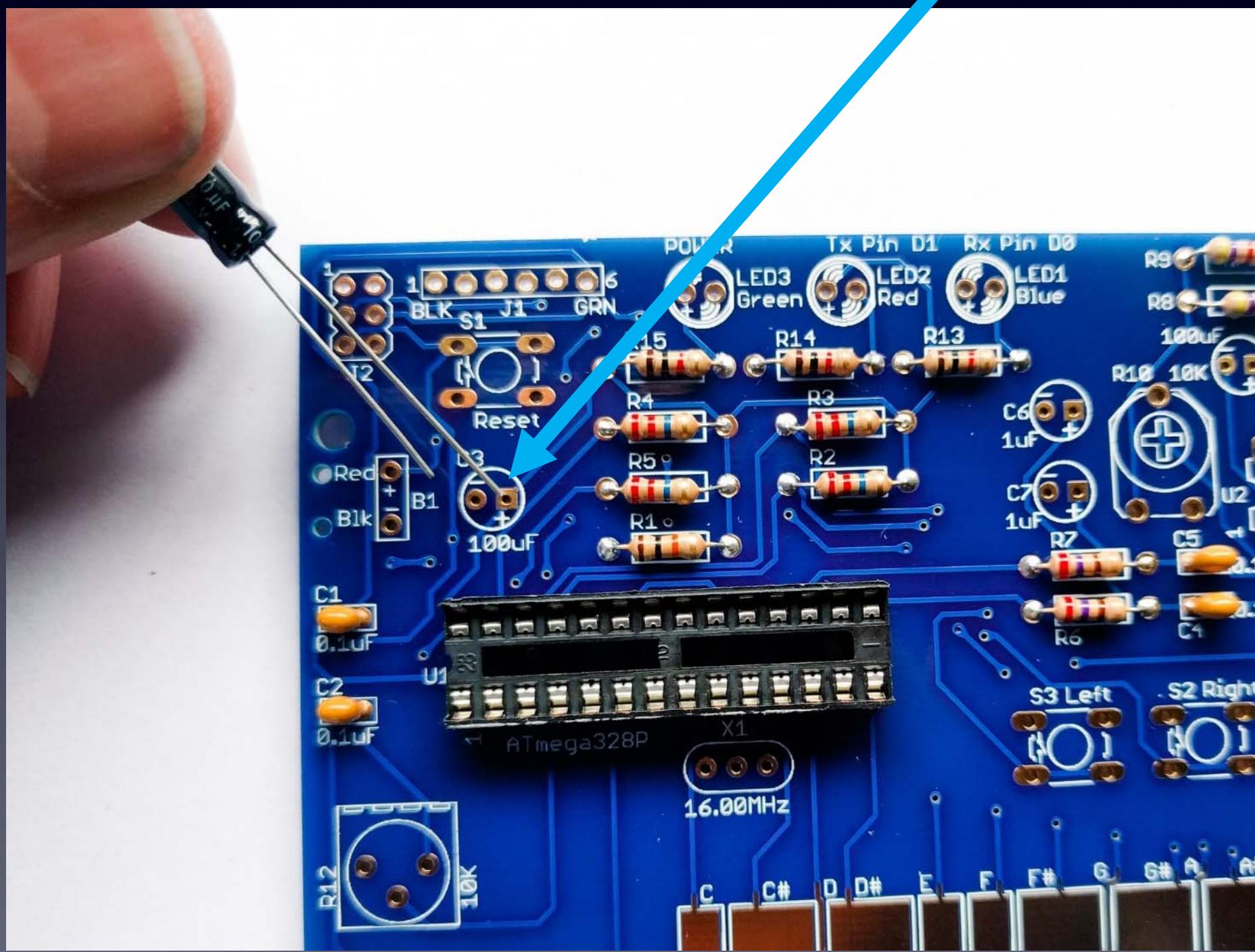
Different than C3, C8 !
C6, C7: 1uF

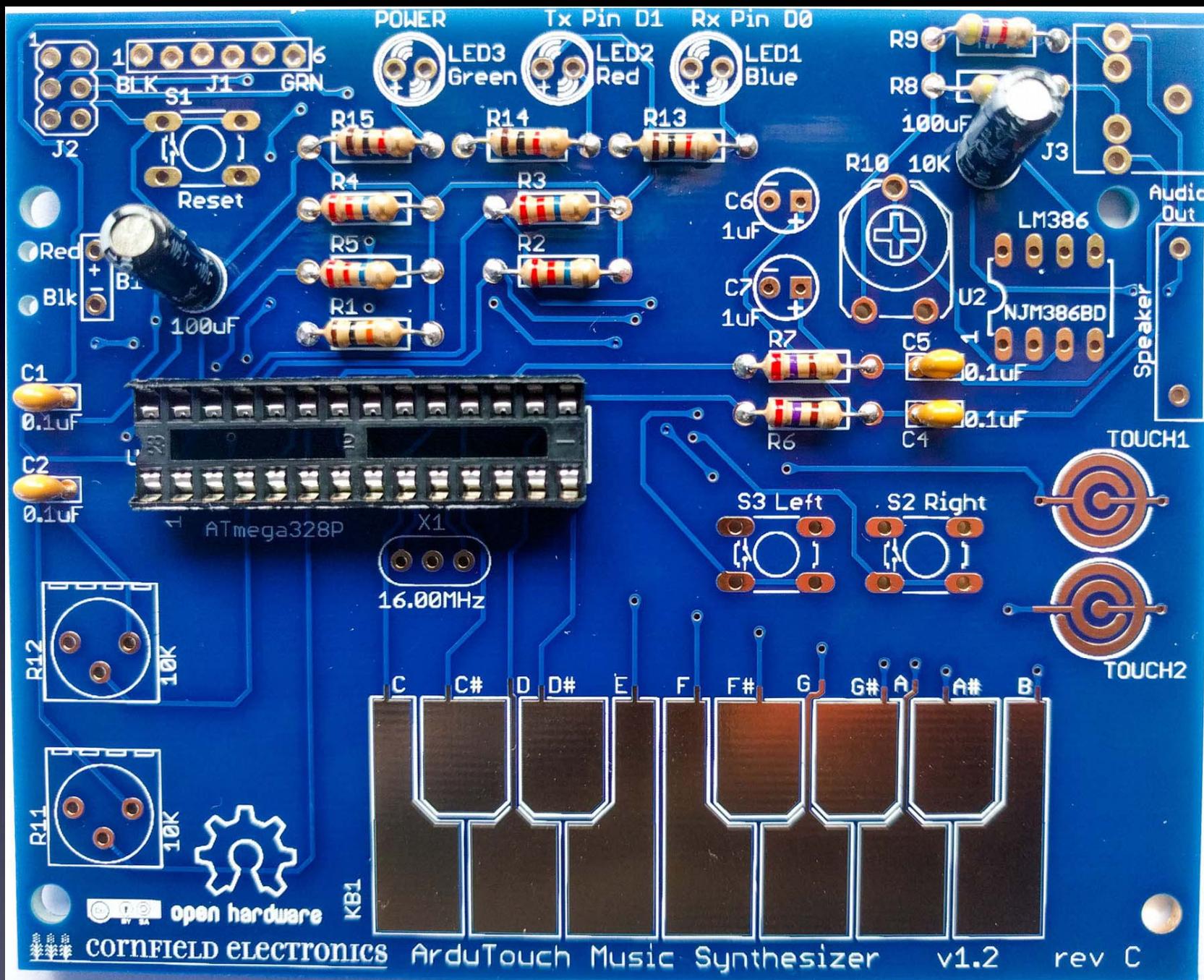


C3, C8: 100uF

C3, C8:
Long Lead “+”

Use 100uF !!

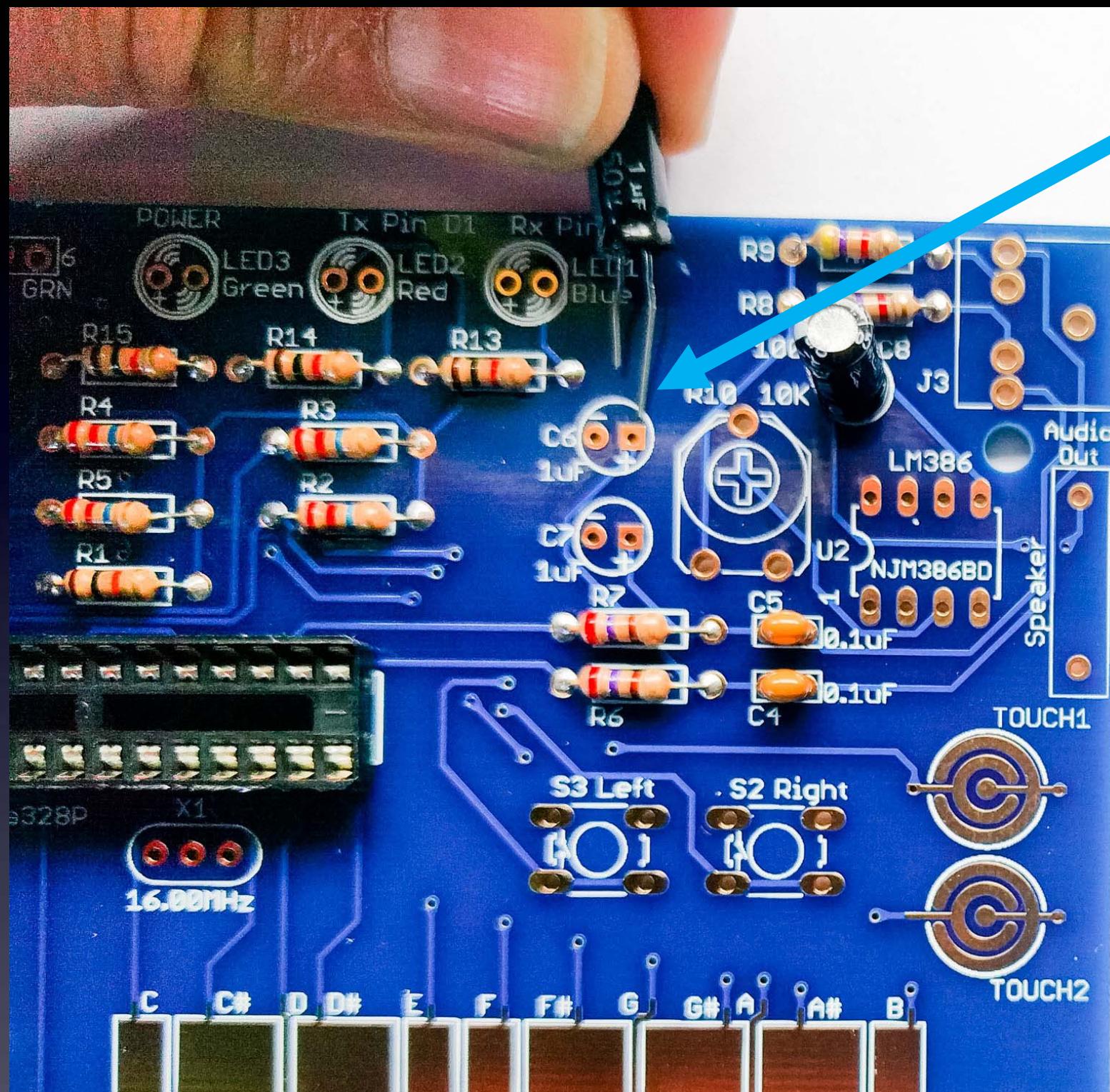




C3, C8: 100uF – soldered to board

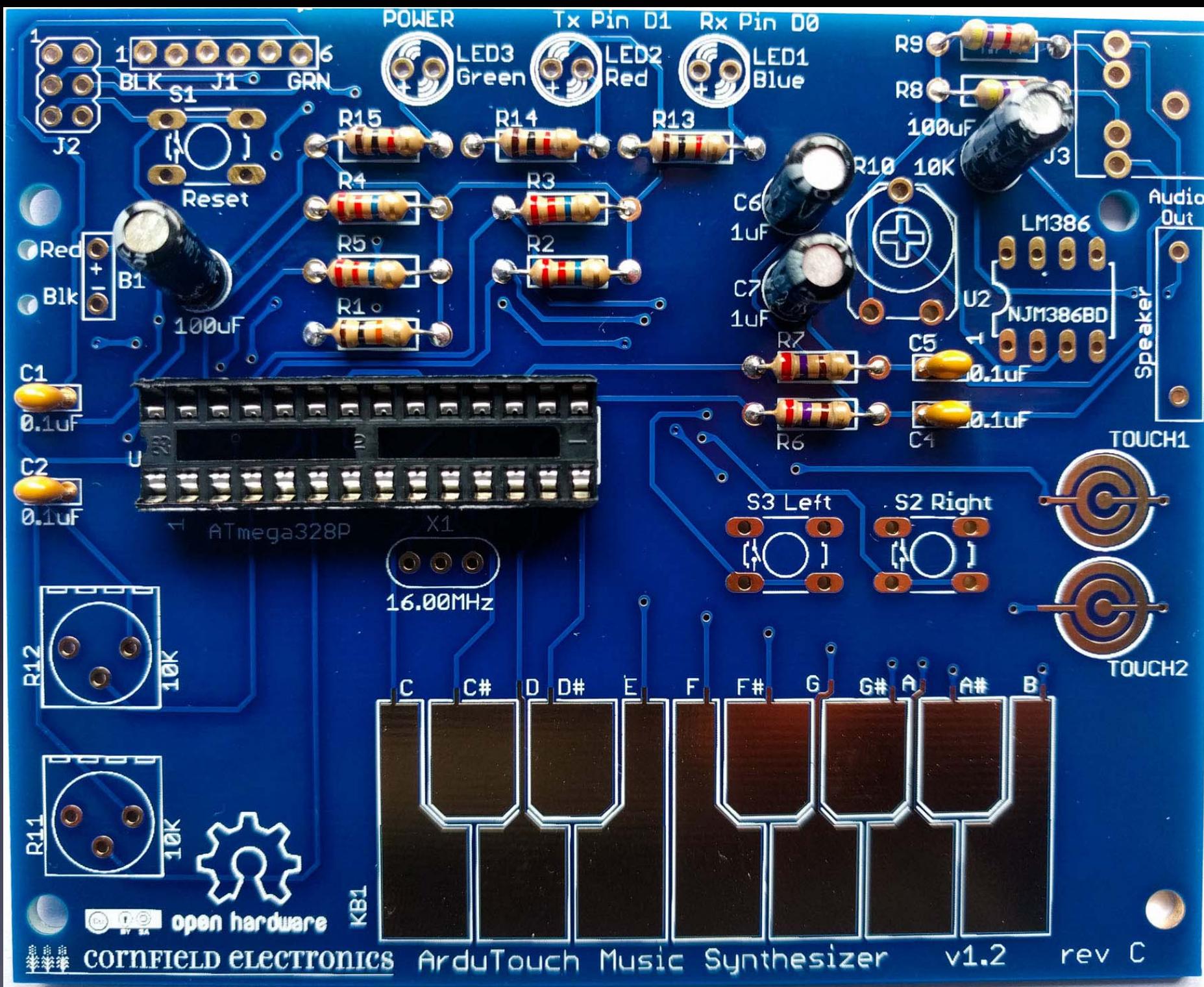


C6, C7: 1uF



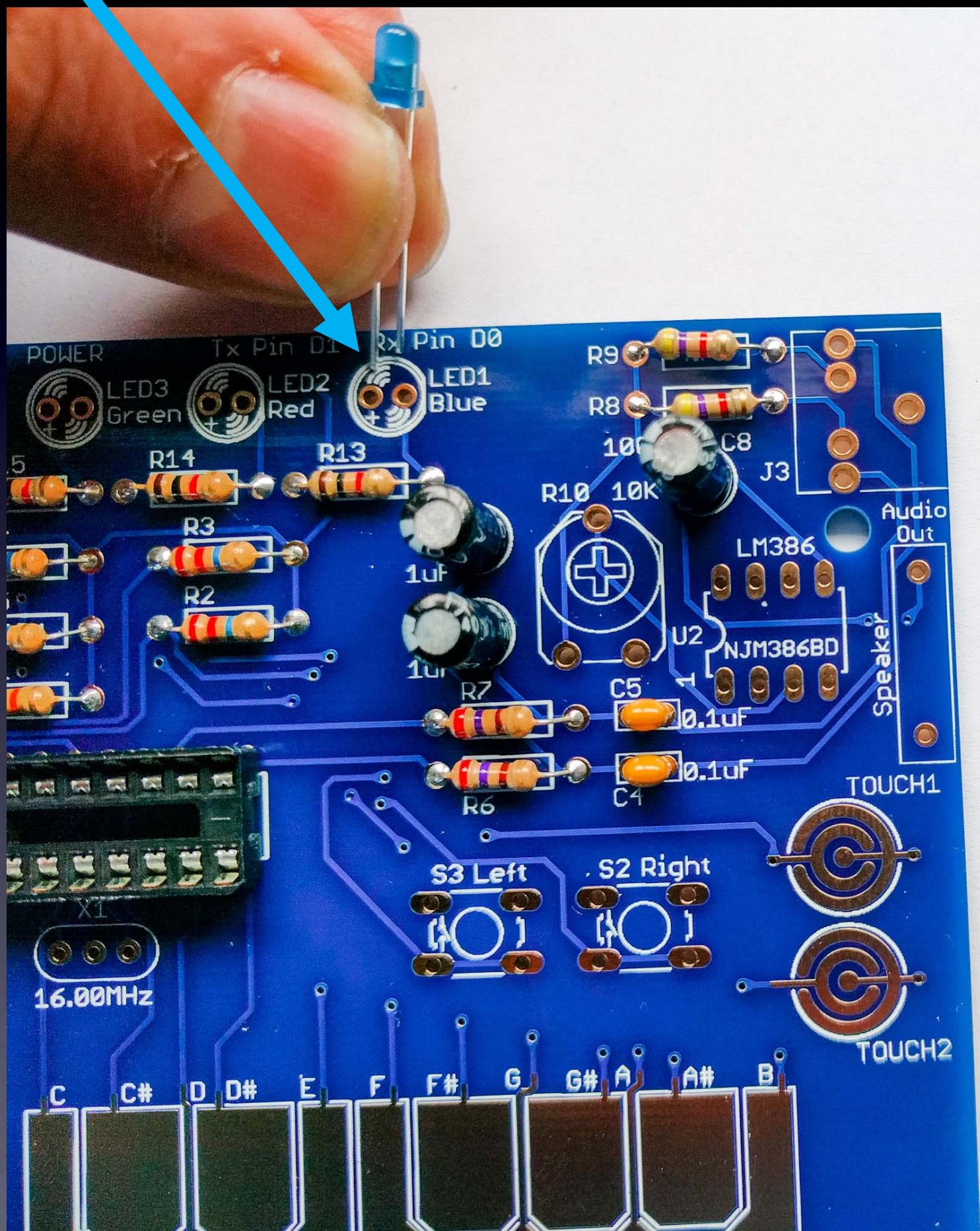
C6, C7:
Long Lead “+”

Use 1uF !!



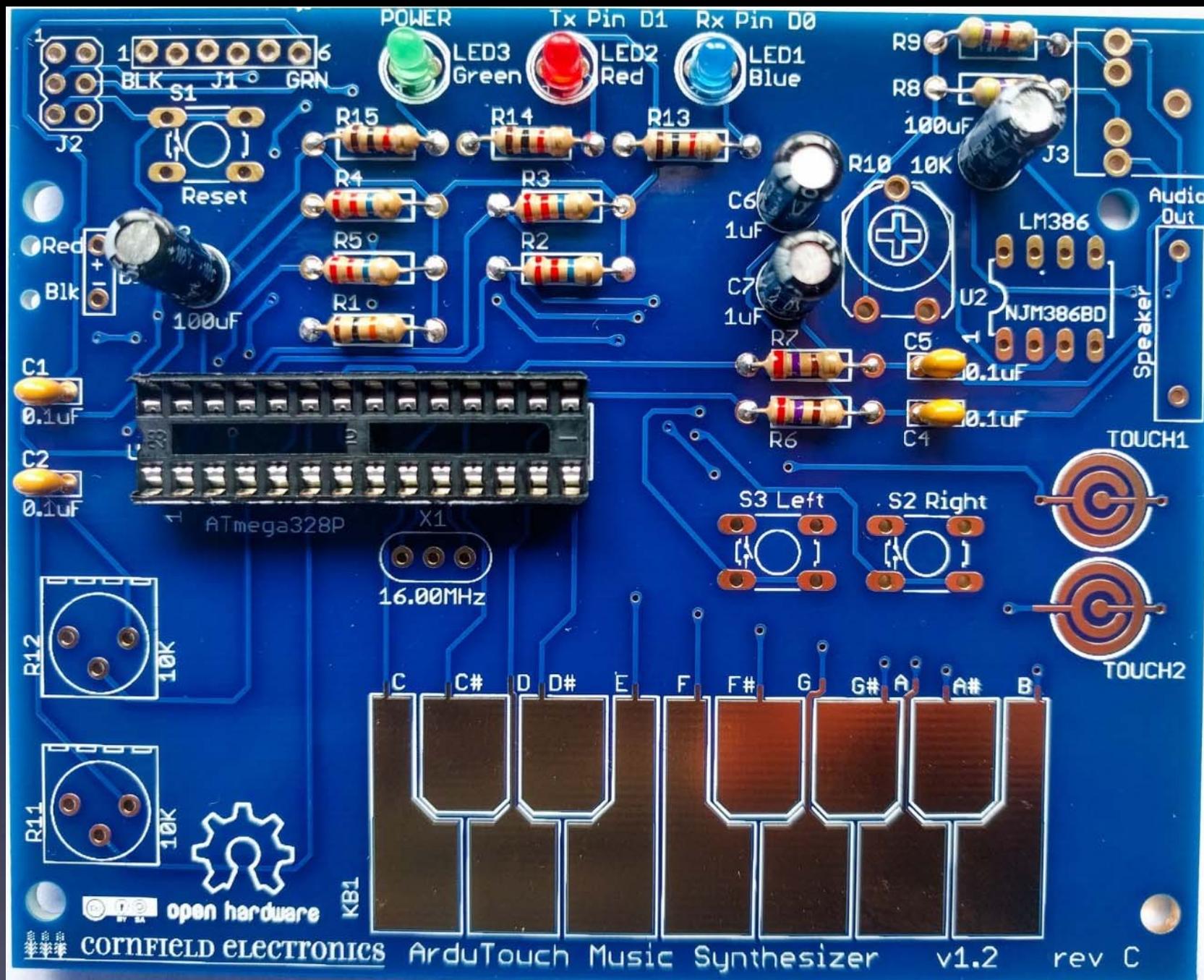
C6, C7: 1uF – soldered to board

LED1, LED2, LED3: Long Lead “+”



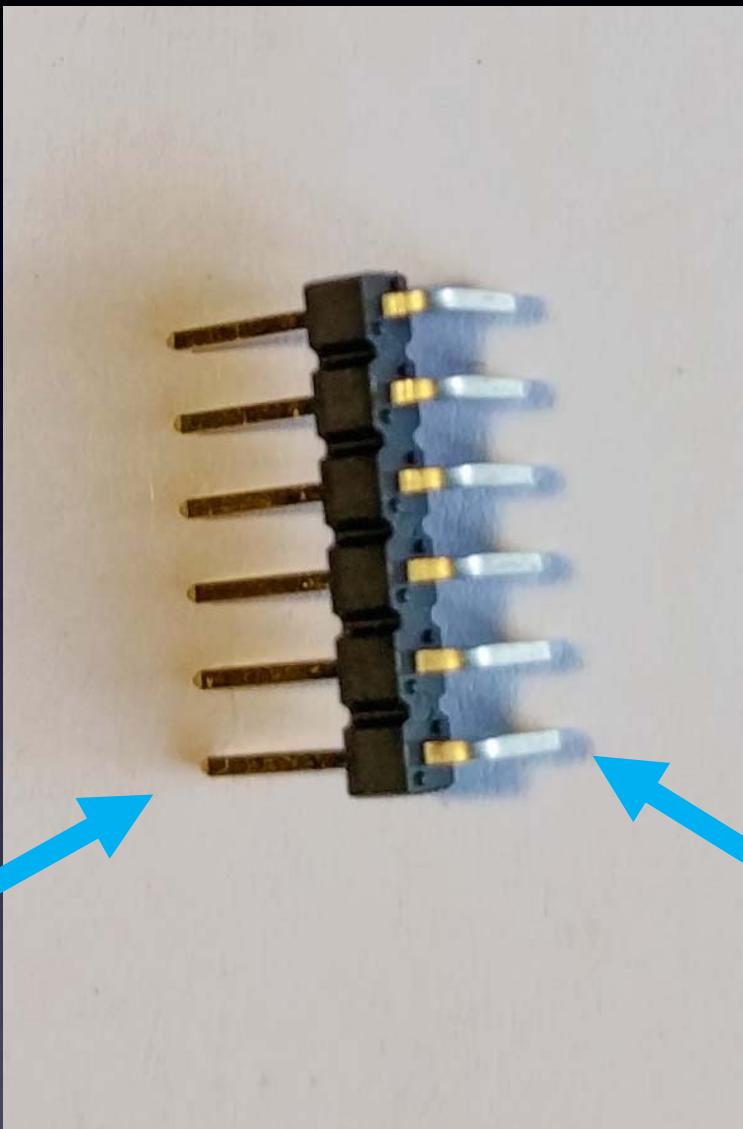
Save
these leads

We'll use them for the speaker



LED1, LED2, LED3

Green, Red, Blue – soldered to board



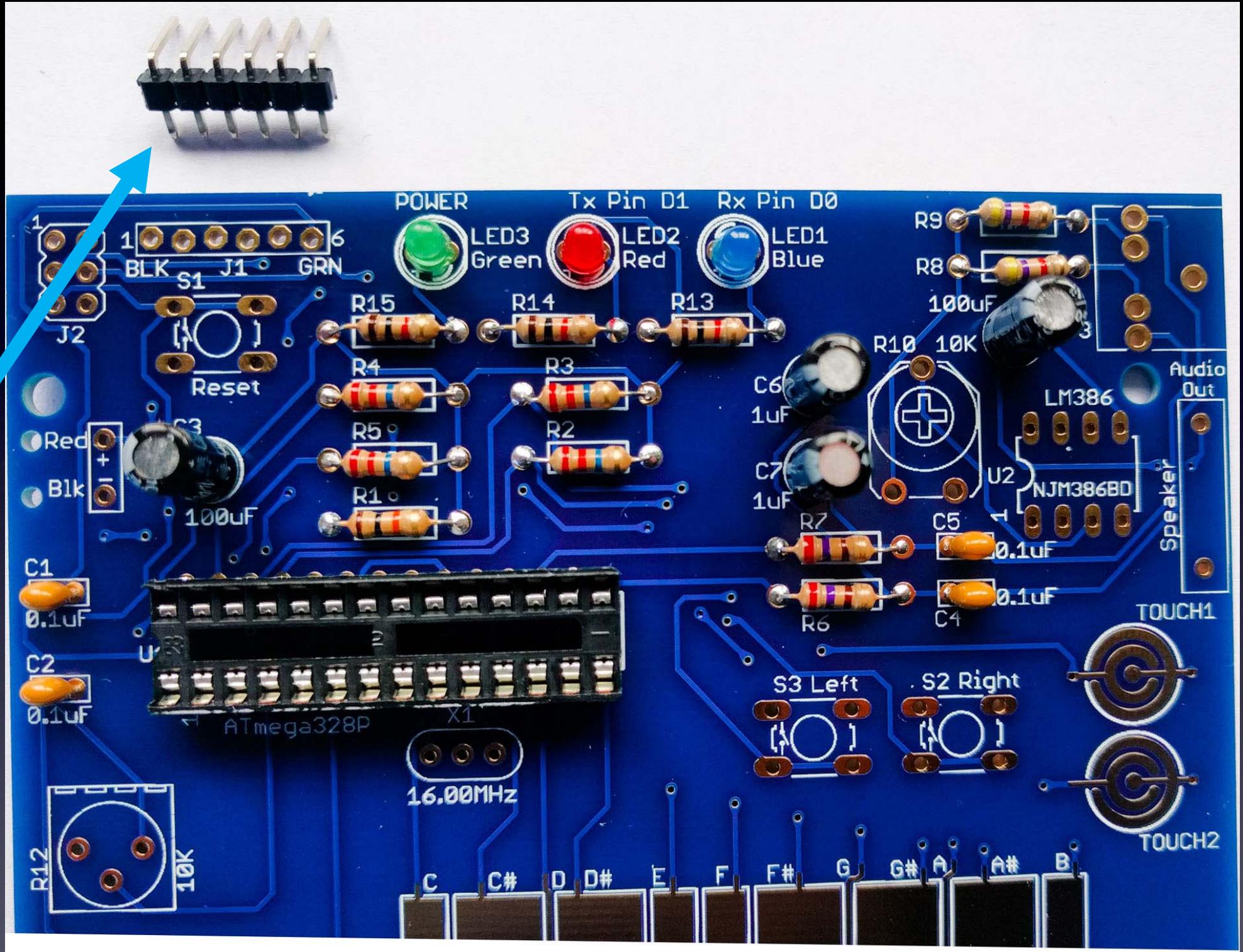
long leads

short leads

J1

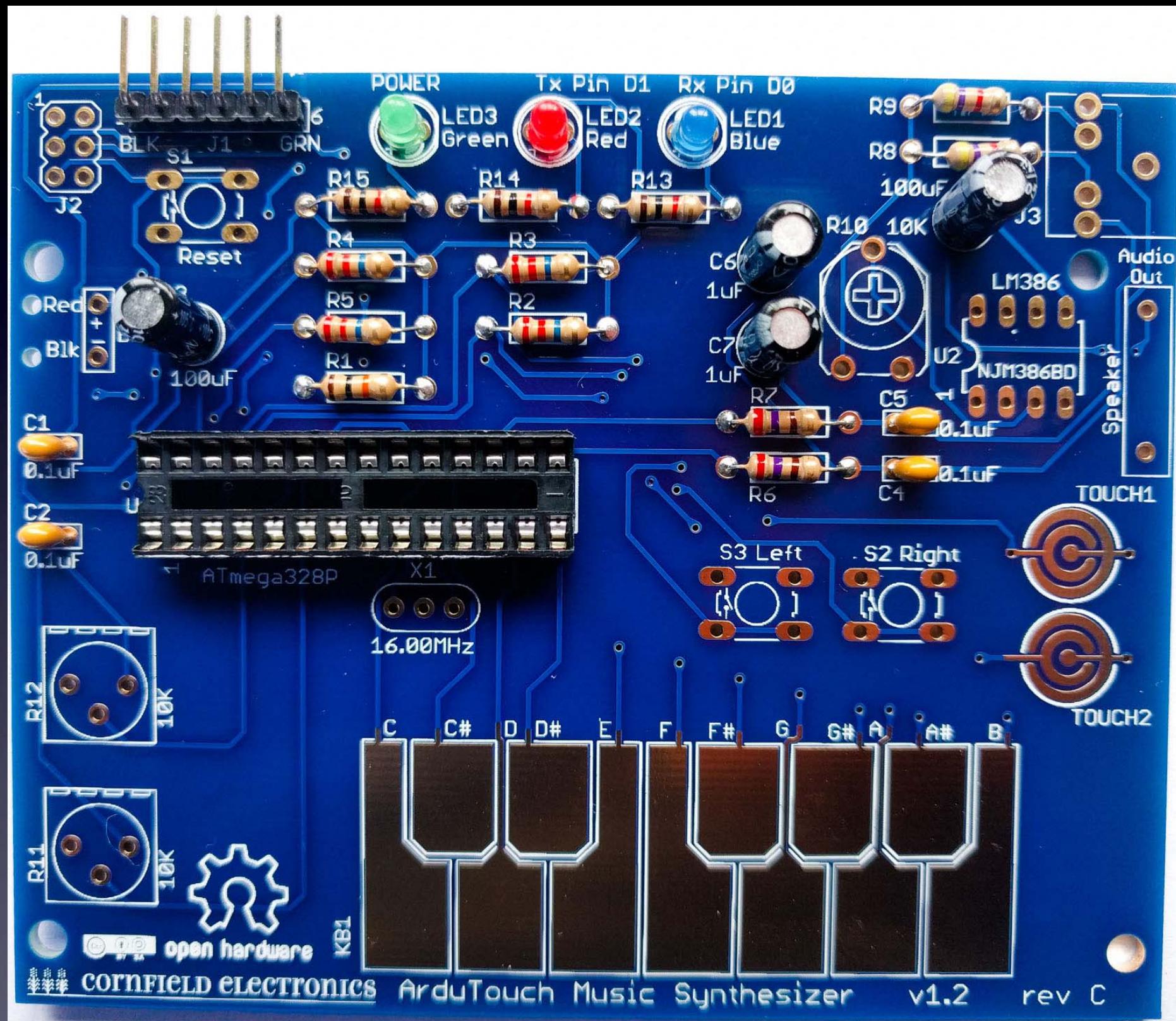
Short leads into board

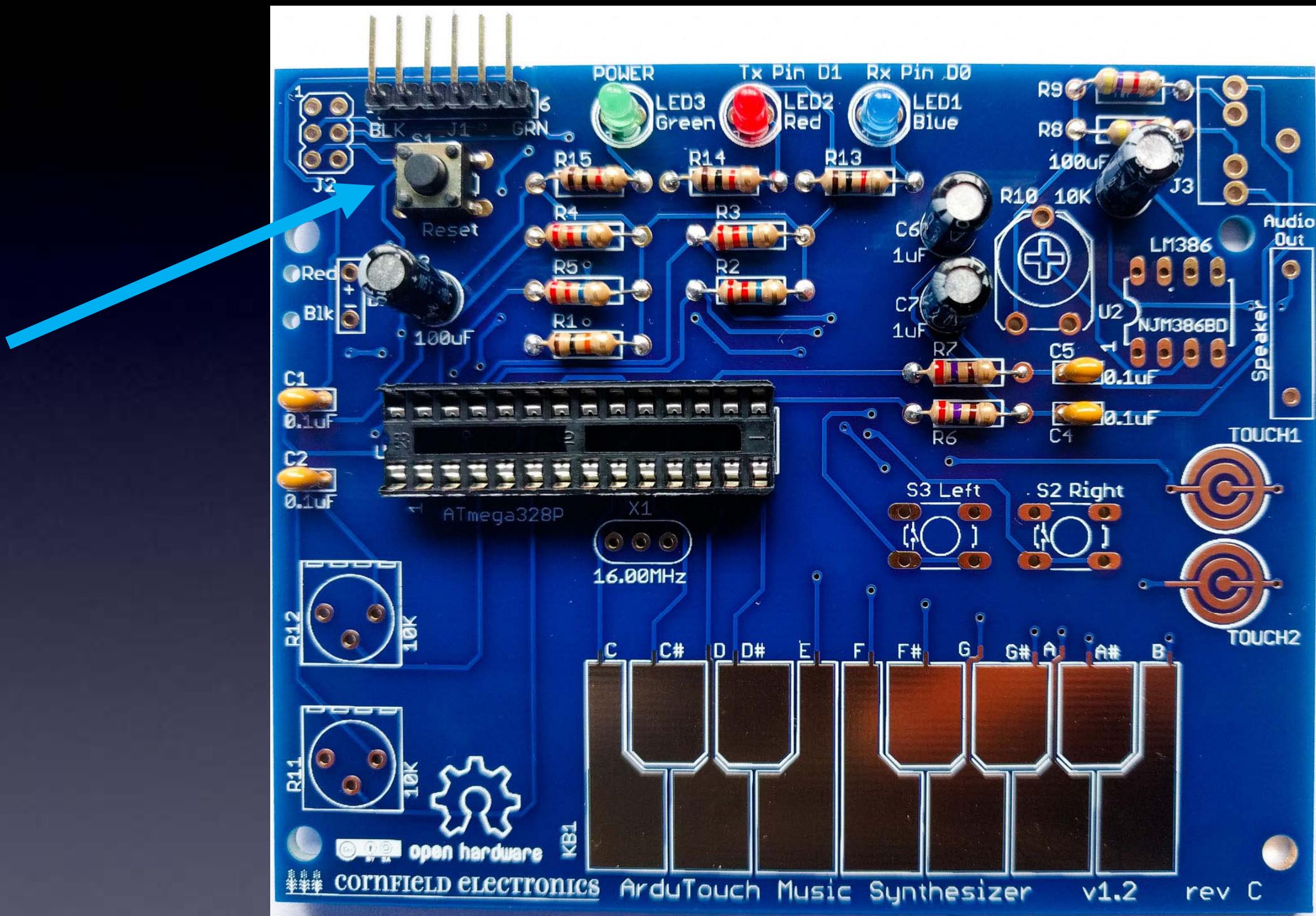
J1
short leads
go into the board



→ long leads sticking out from
board

J1



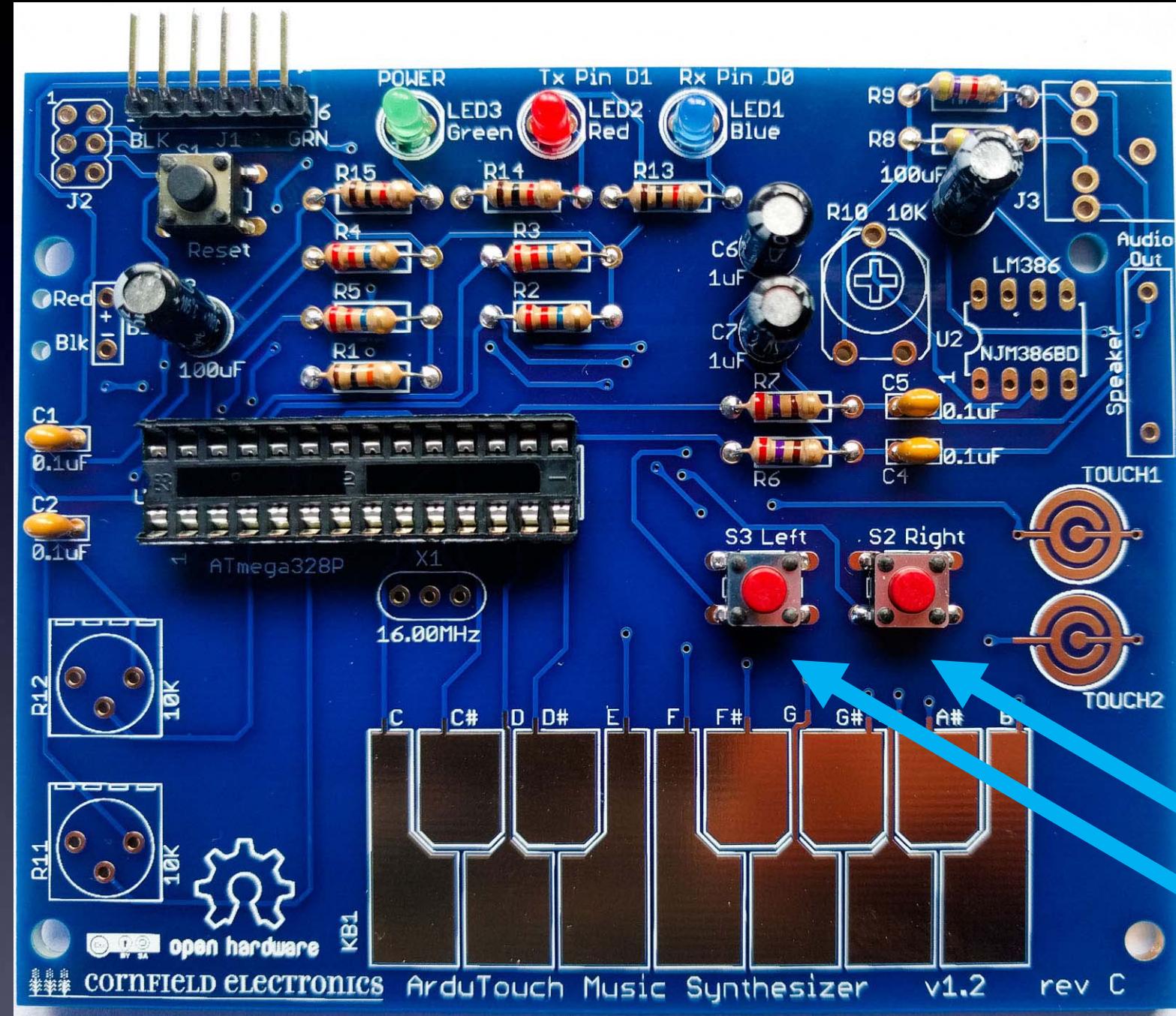


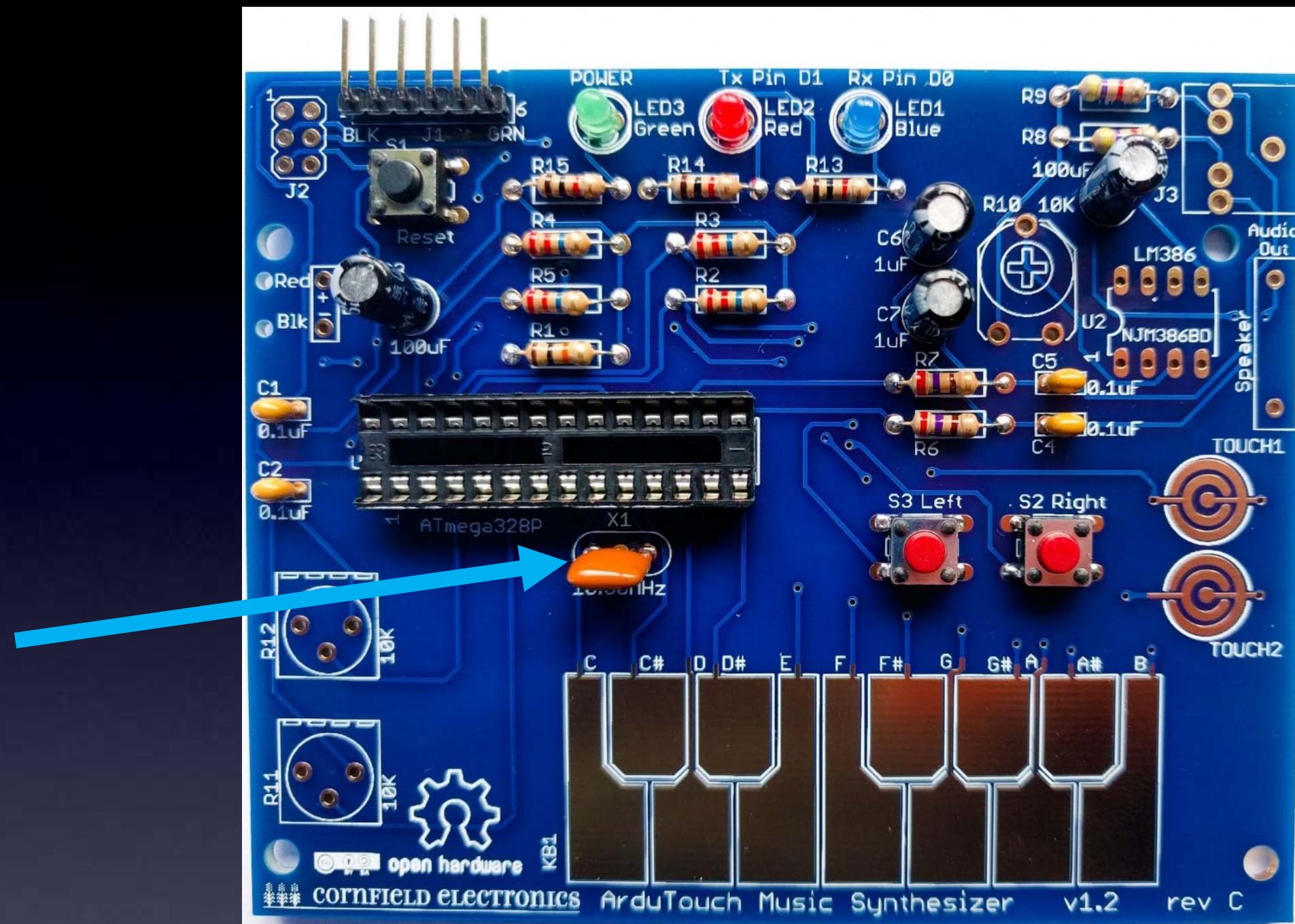
S1: black Reset button

Note: The color of this switch is not important (some kits may have different colors).

S2, S3: Red buttons

Note: The color of these switches is not important (some kits may have different colors).





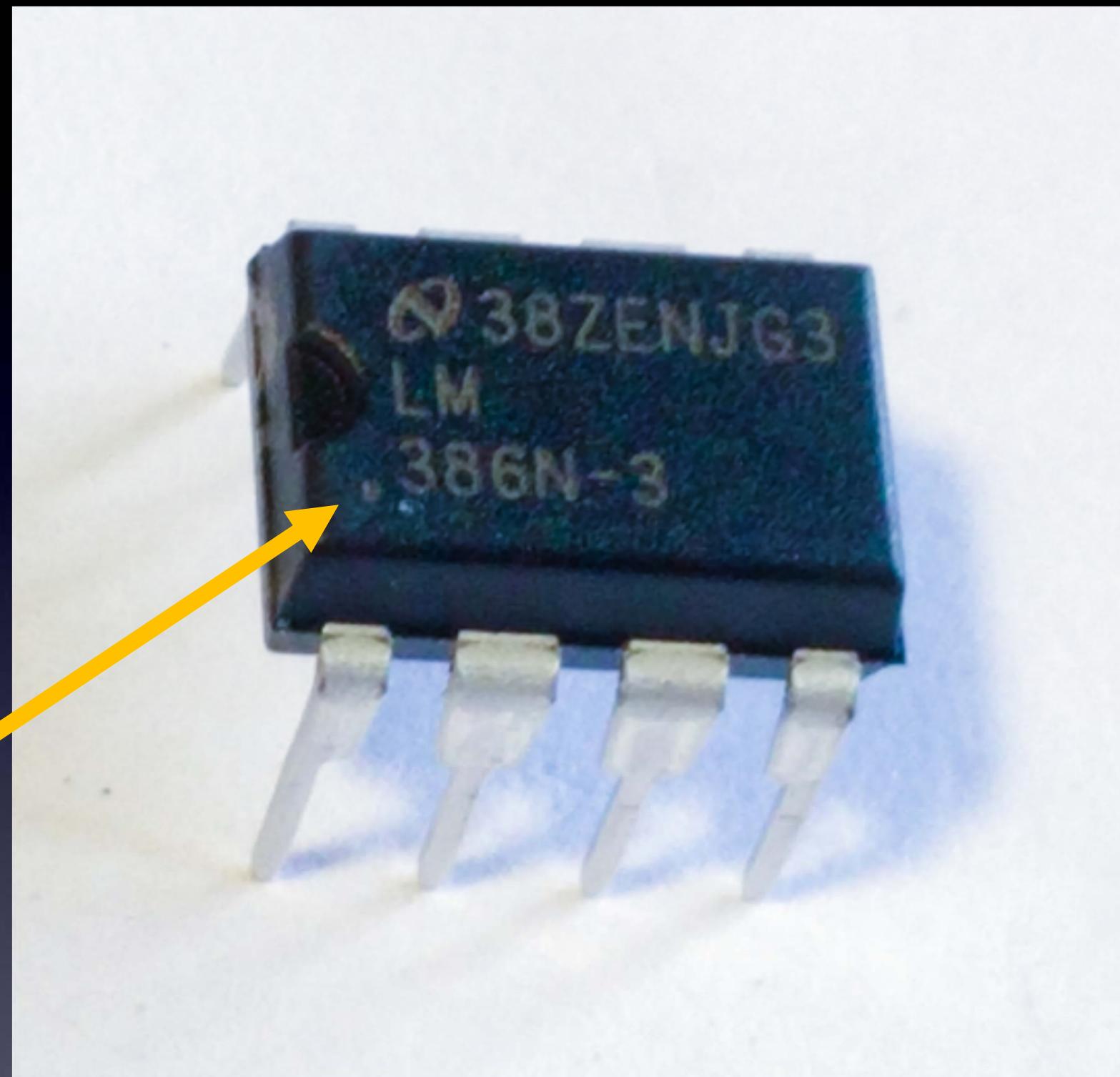
X1

The orientation of X1 does not matter.

Note: X1 may be yellow or blue.

U2

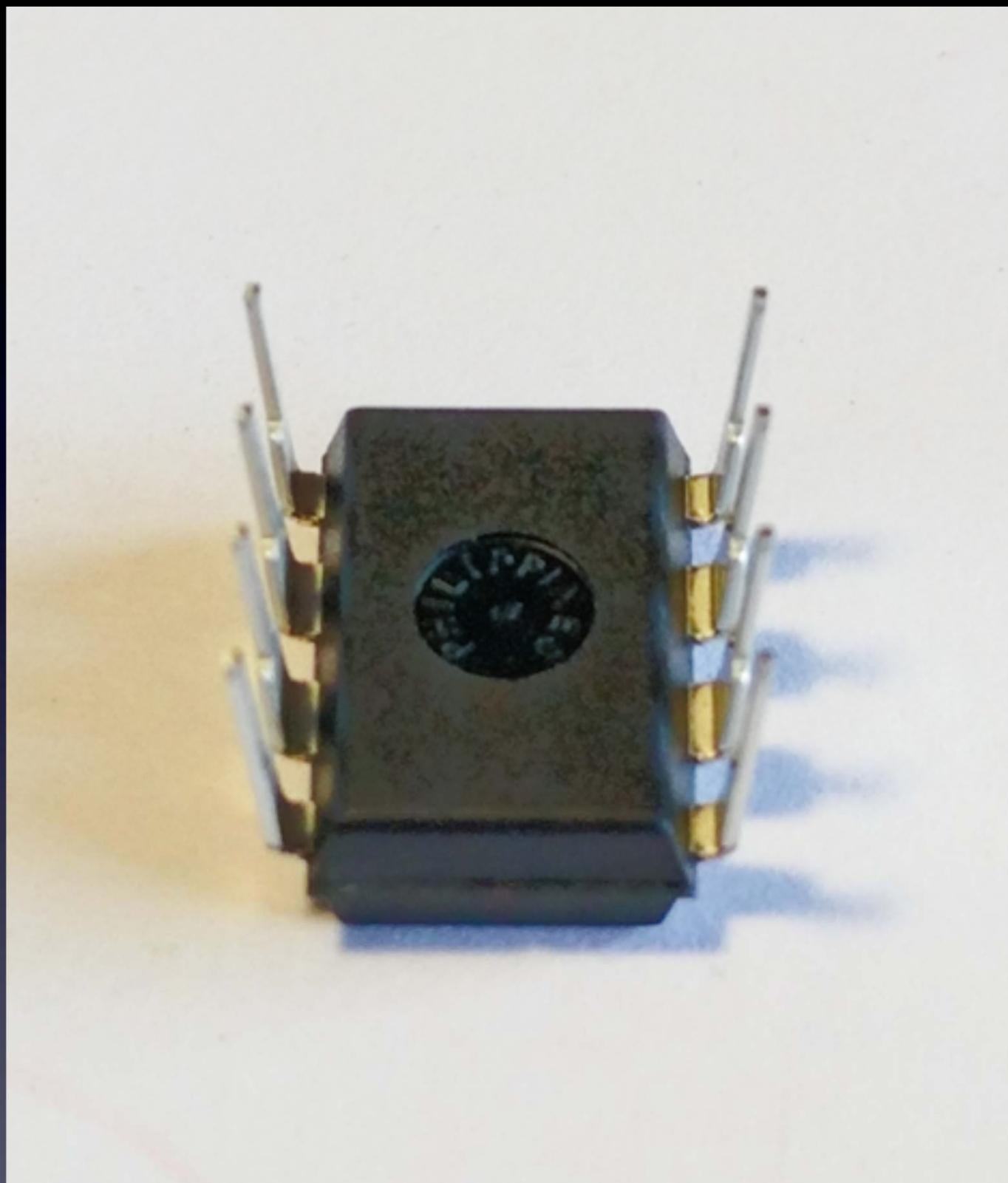
Indented black dot
Pin 1



Note: Your chip may be marked differently, but “386” will be printed on it somewhere.

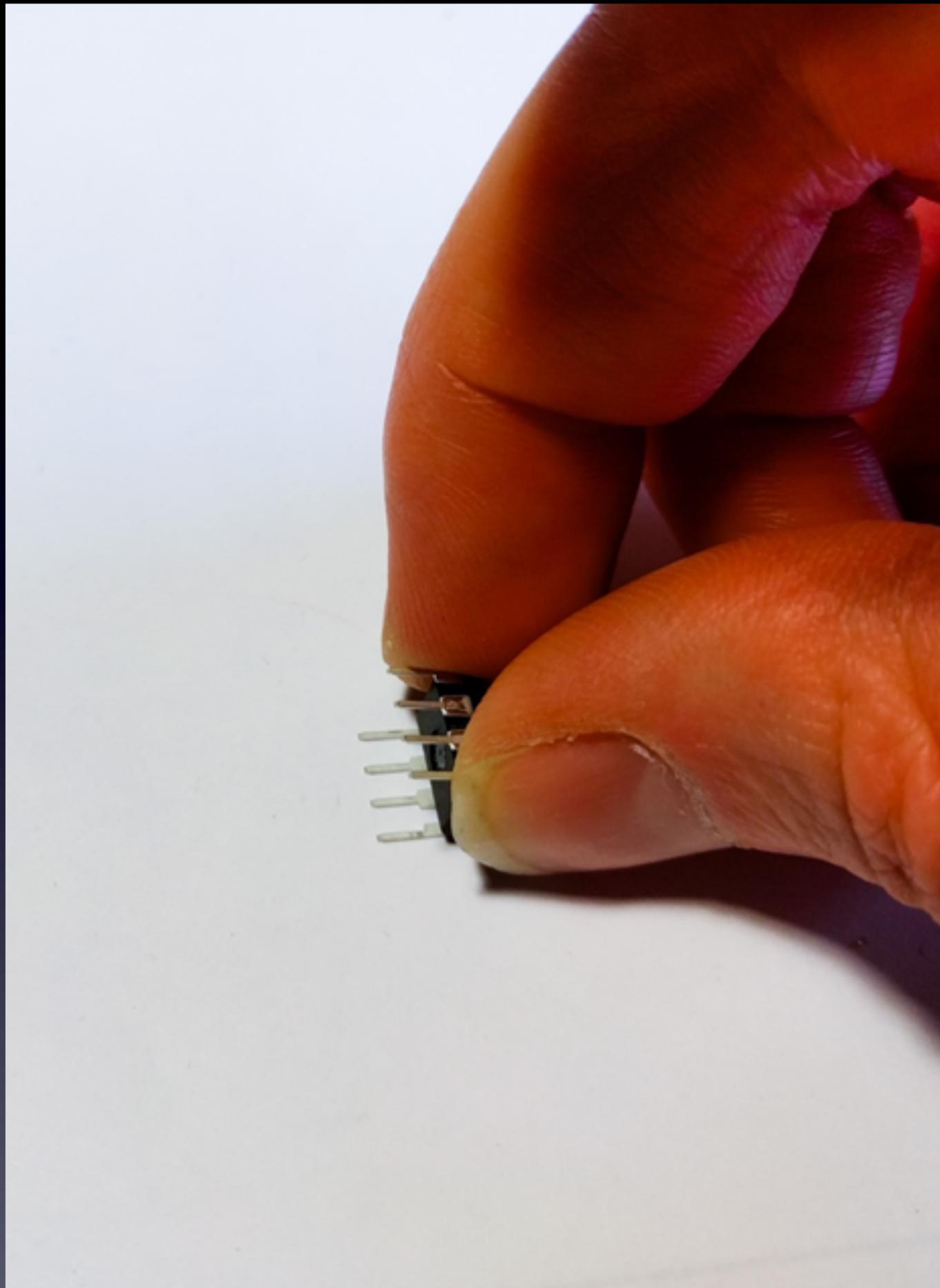
Note: Your chip may or may not have the indented half-moon at the left,
it may have a black indented dot at the lower-left corner showing Pin 1.

U2



When chips are new,
their pins are bent out.

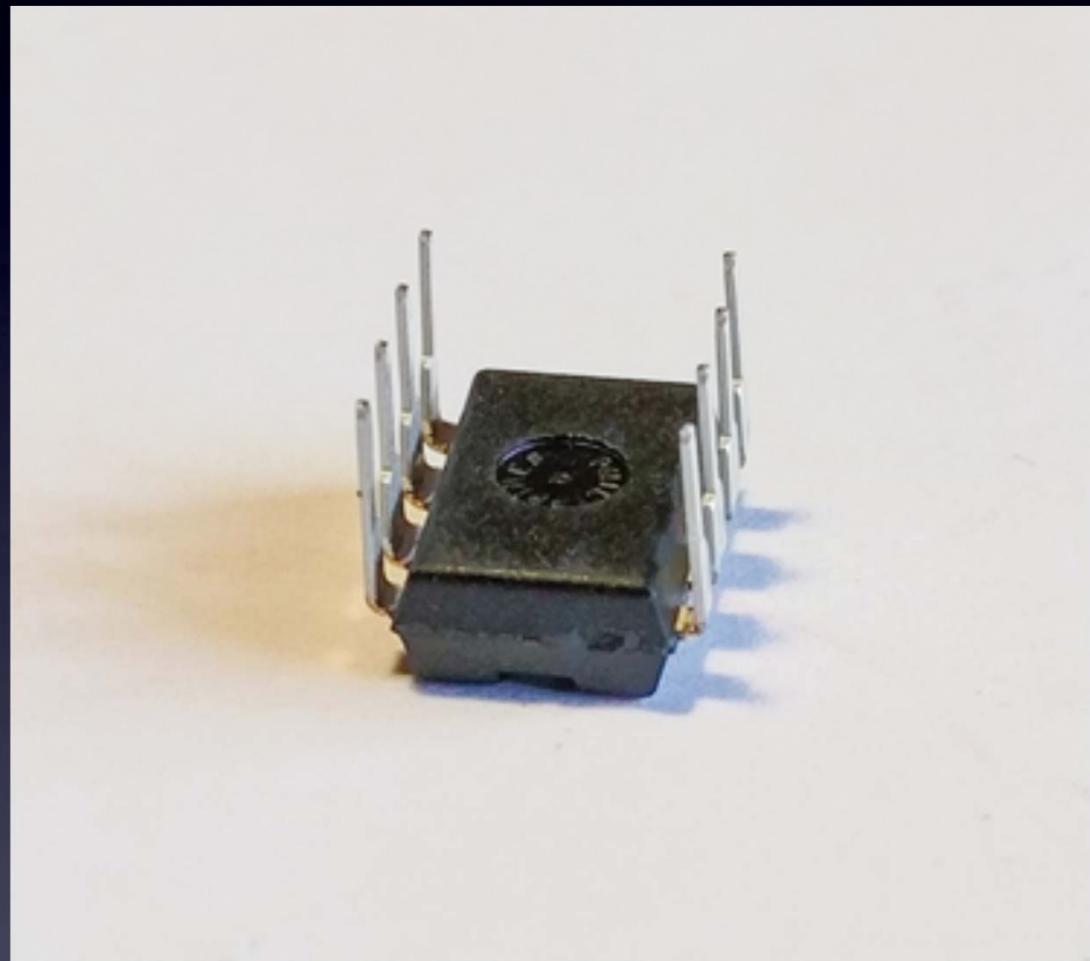
U2

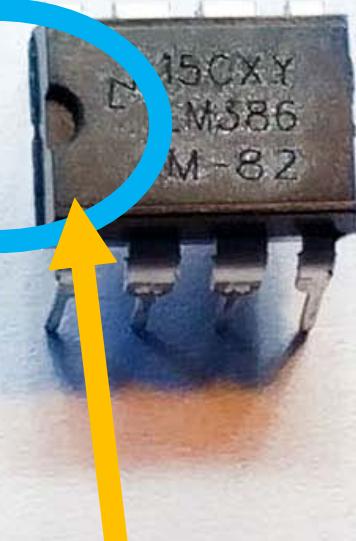
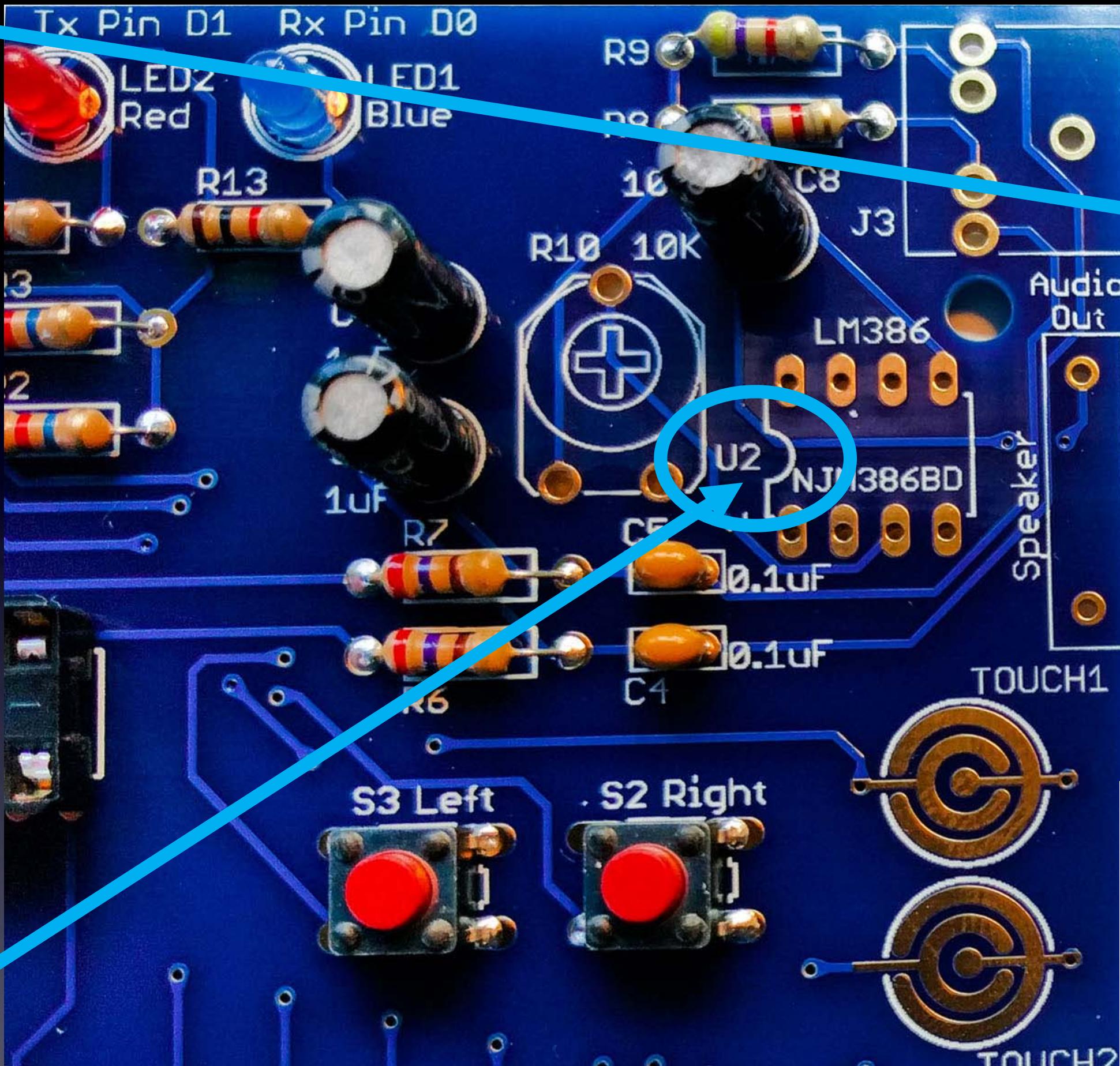


We need the pins bent straight and parallel.
Use your work table to (gently) bend the leads.

U2

Gently
bend leads
so they're straight
and parallel



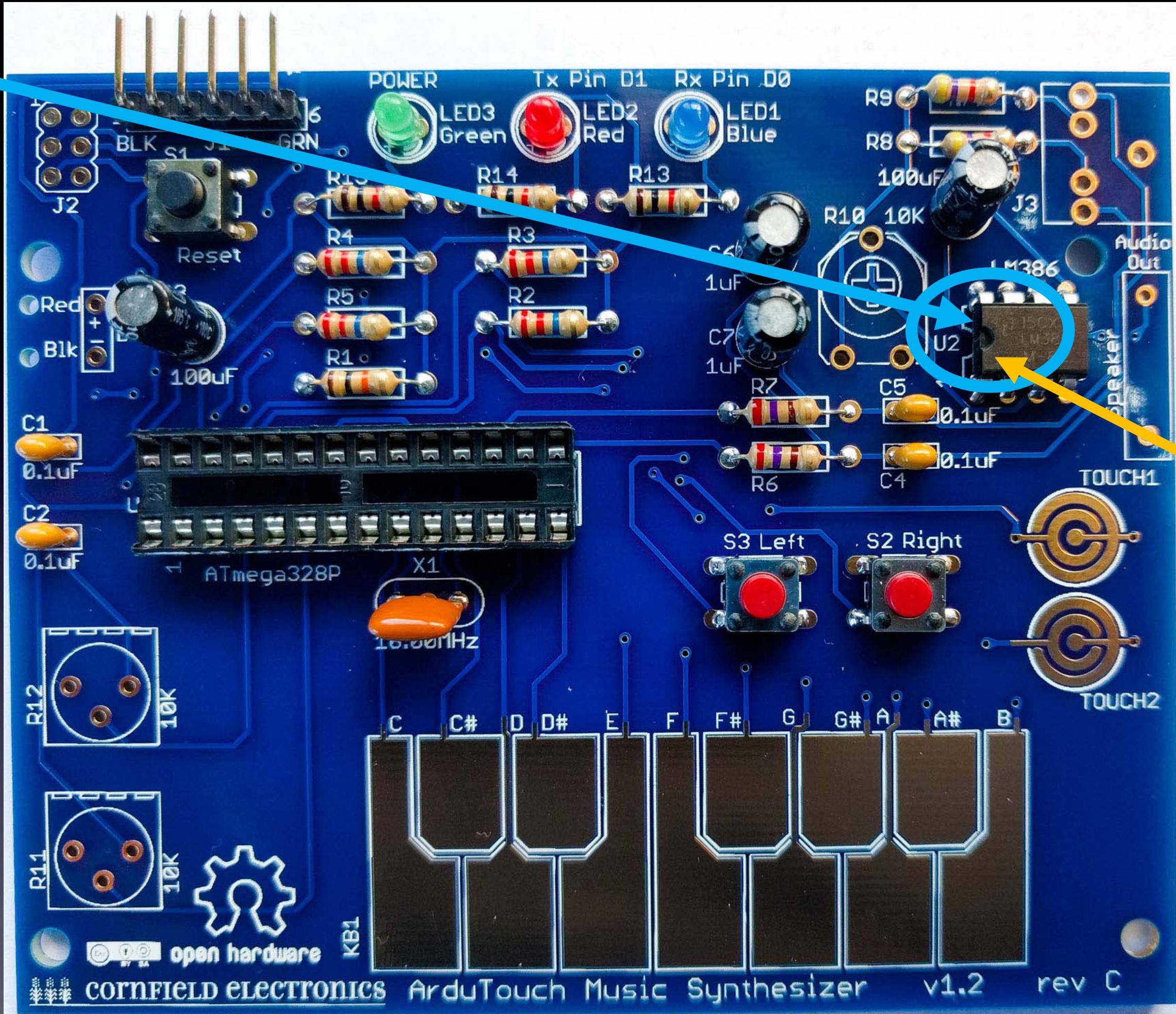


Indented black dot
Pin 1

**proper
orientation**

Note: Your chip may or may not have the indented half-moon at the left, it may have a black indented dot at the lower-left corner showing Pin 1.

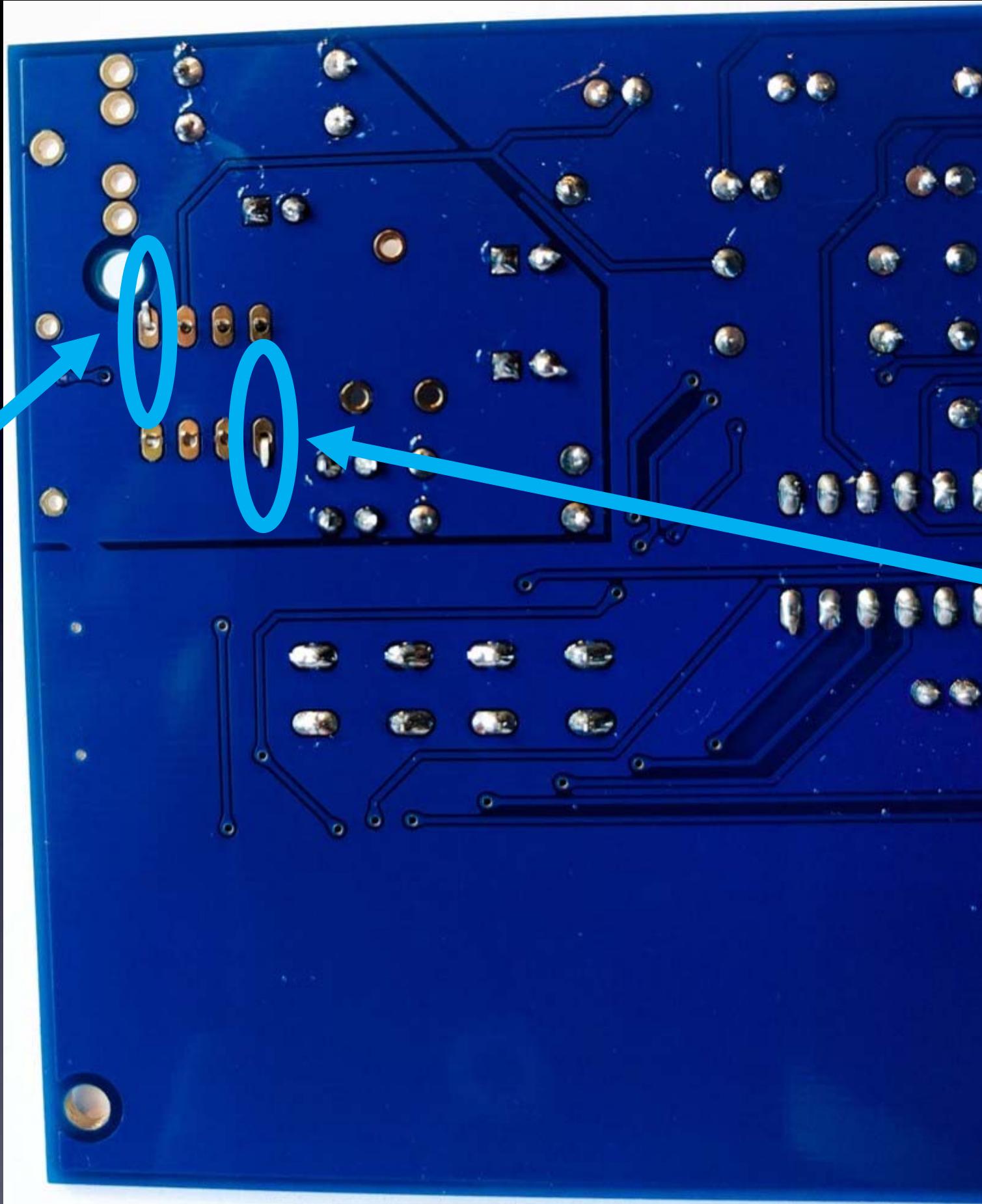
U2: audio amp chip



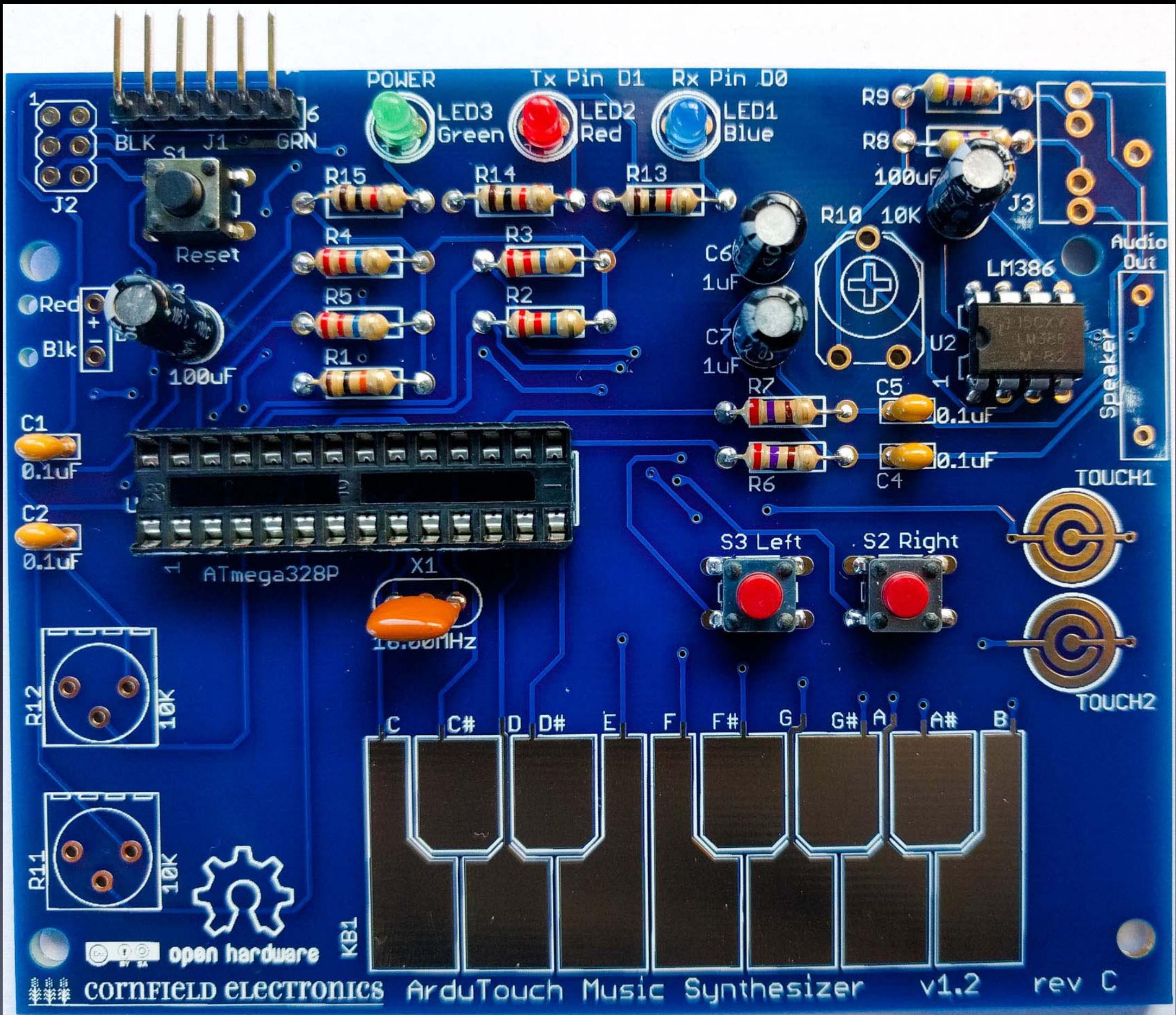
U2: inserted correctly

Indented black dot
Pin 1

U2

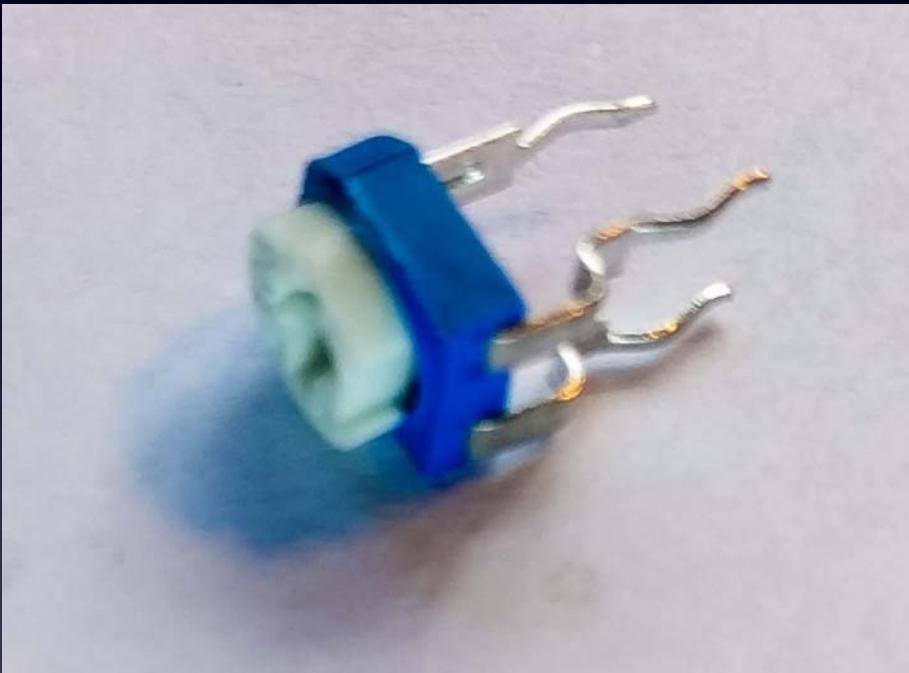


bend pins down on two corners,
and solder all 8 leads to the board



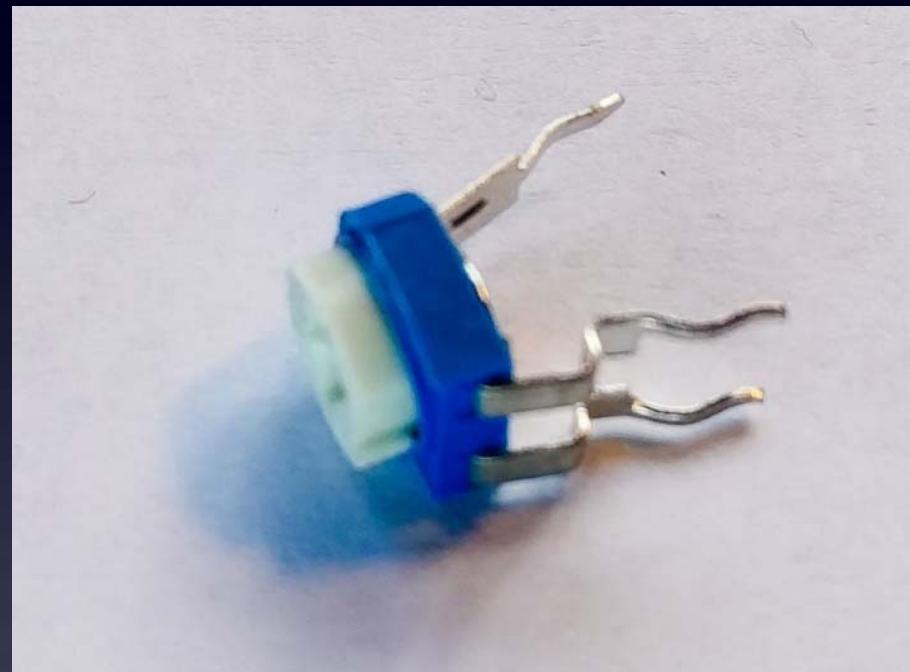
U2 – soldered to board

R10: volume control



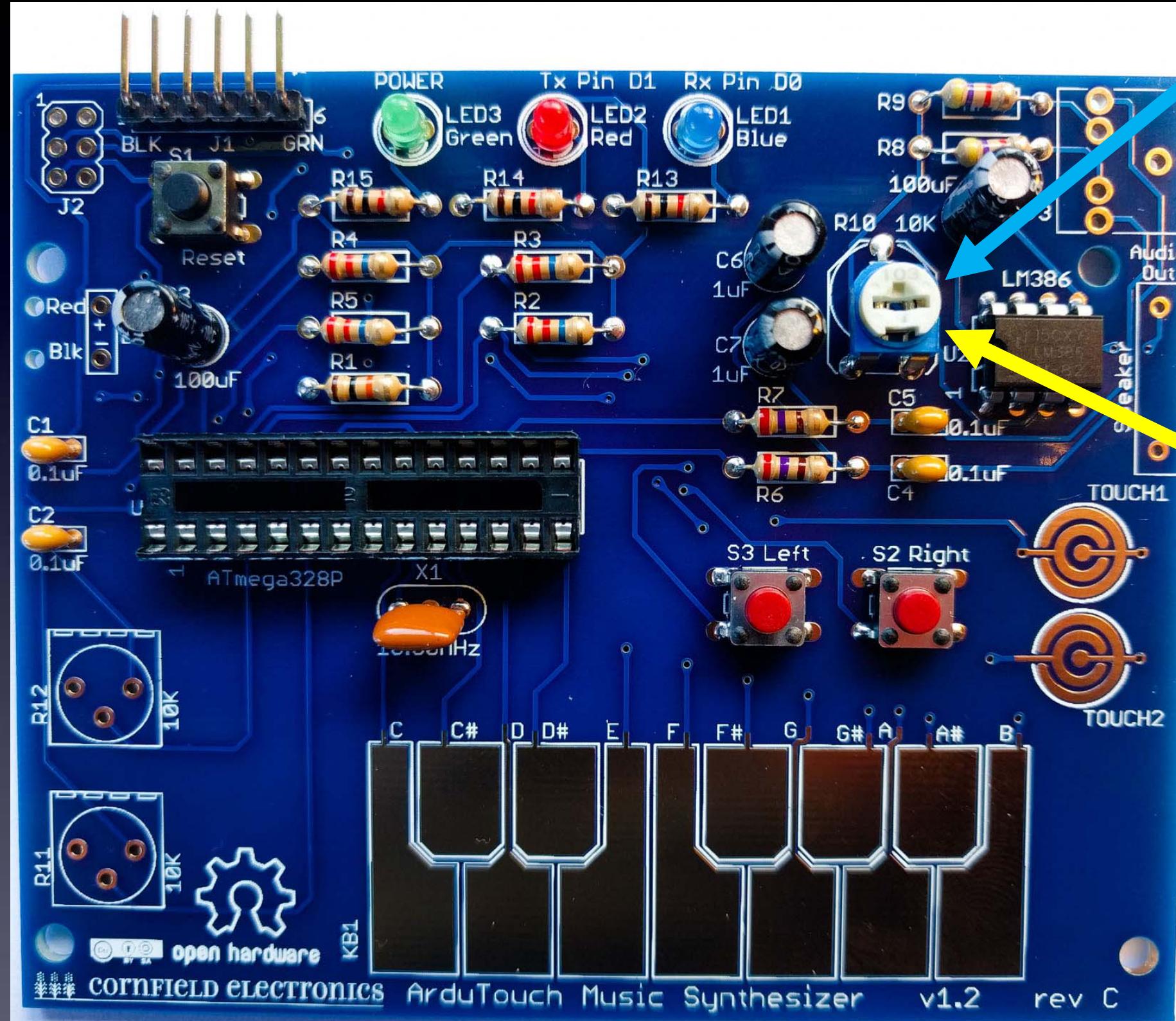
When new, the pins point straight down.

R10: volume control

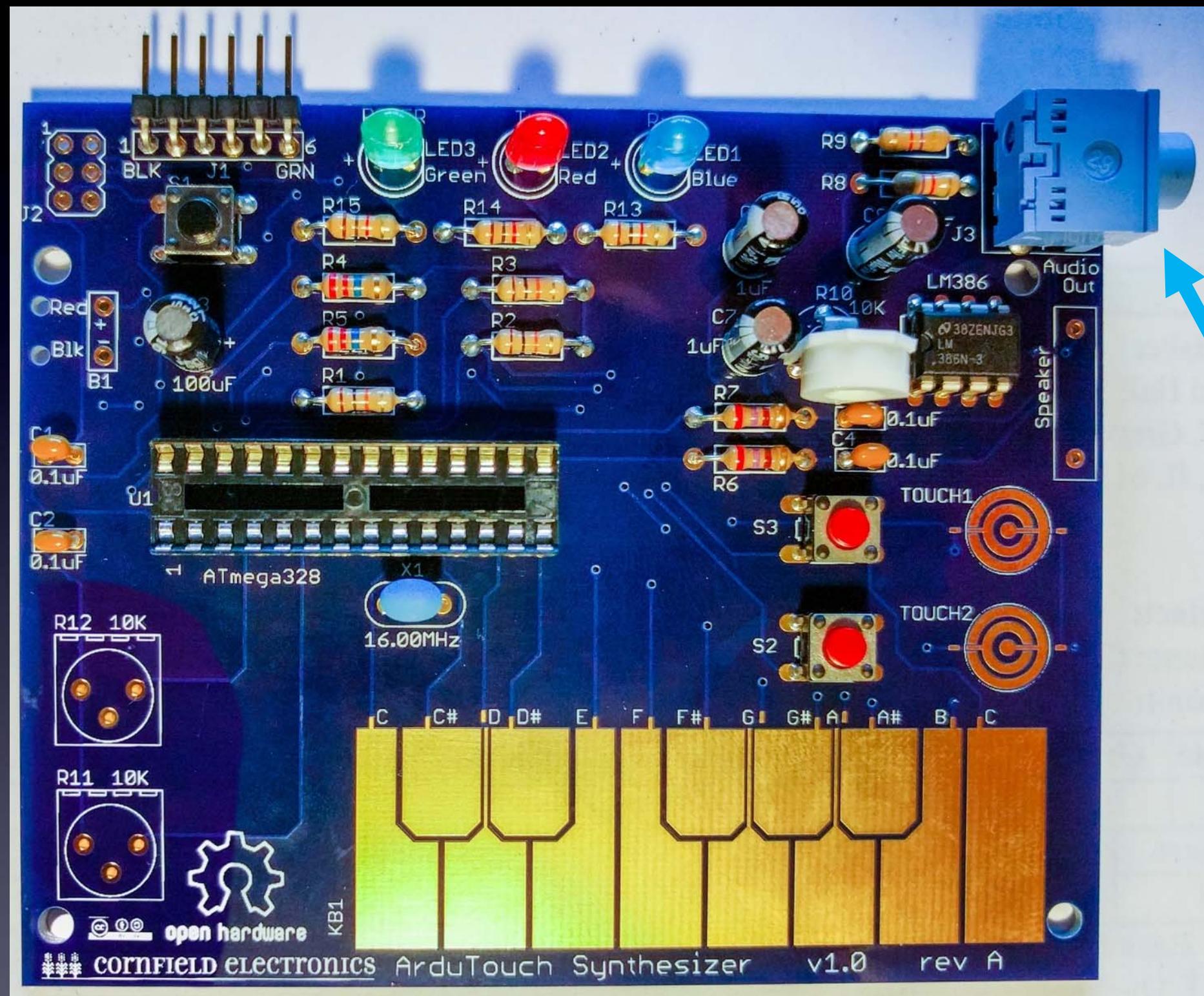


We need to bend them out a little to fit into the board.

R10: volume control



If necessary,
rotate the white top
so that it looks
like this photo
(rotated half-way)



J3: headphone / output jack



U1: microcontroller

U1



**When chips are new,
their pins are bent out.**

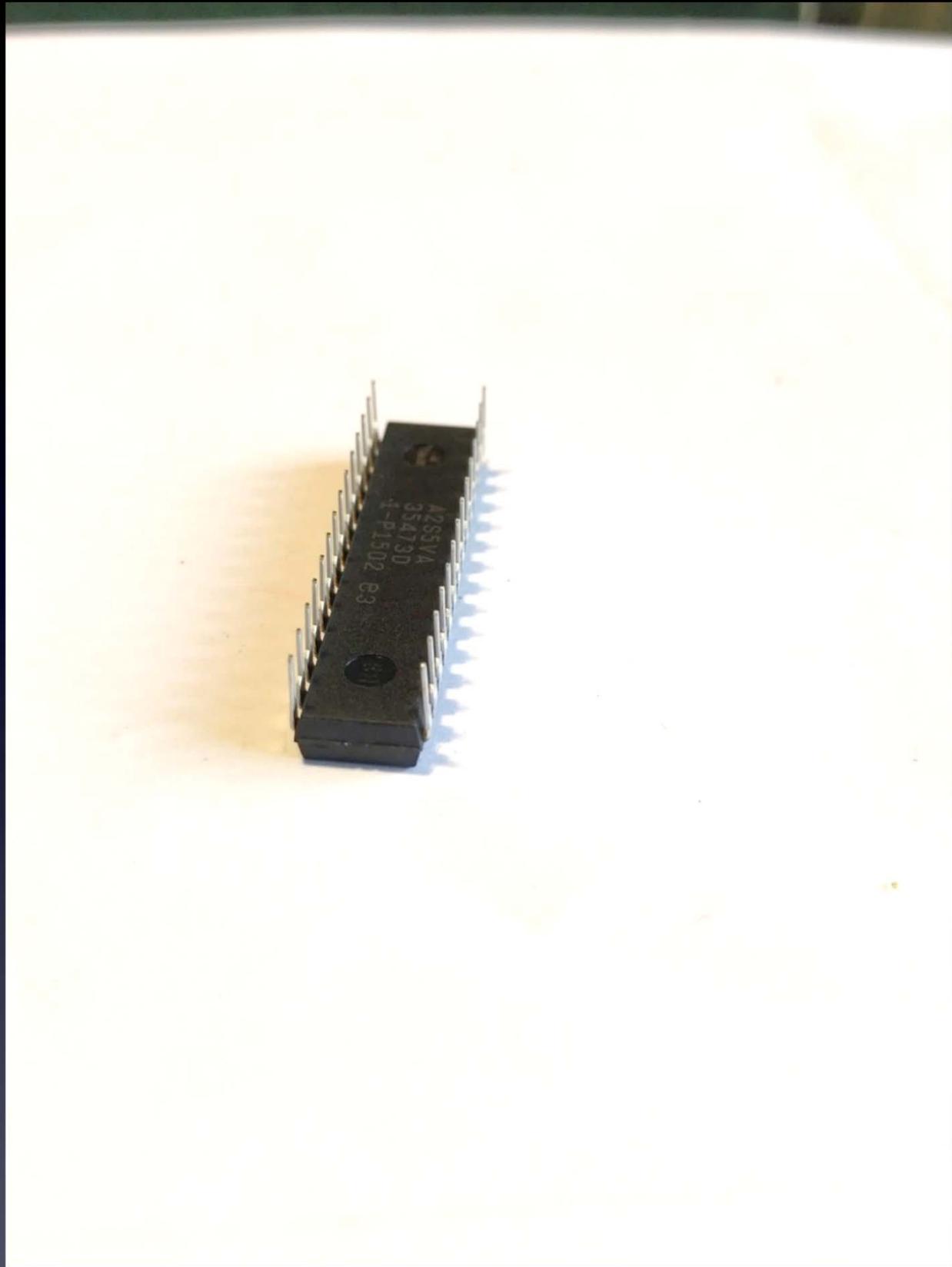
Note: Your kit's U1 chip may or may not have its pins already bent straight and parallel. If not, you need to bend them, as shown in the next picture.

U1

Note: Your kit's U1 chip may or may not have its pins already bent straight and parallel.
If not, you need to bend them, as shown in this picture.

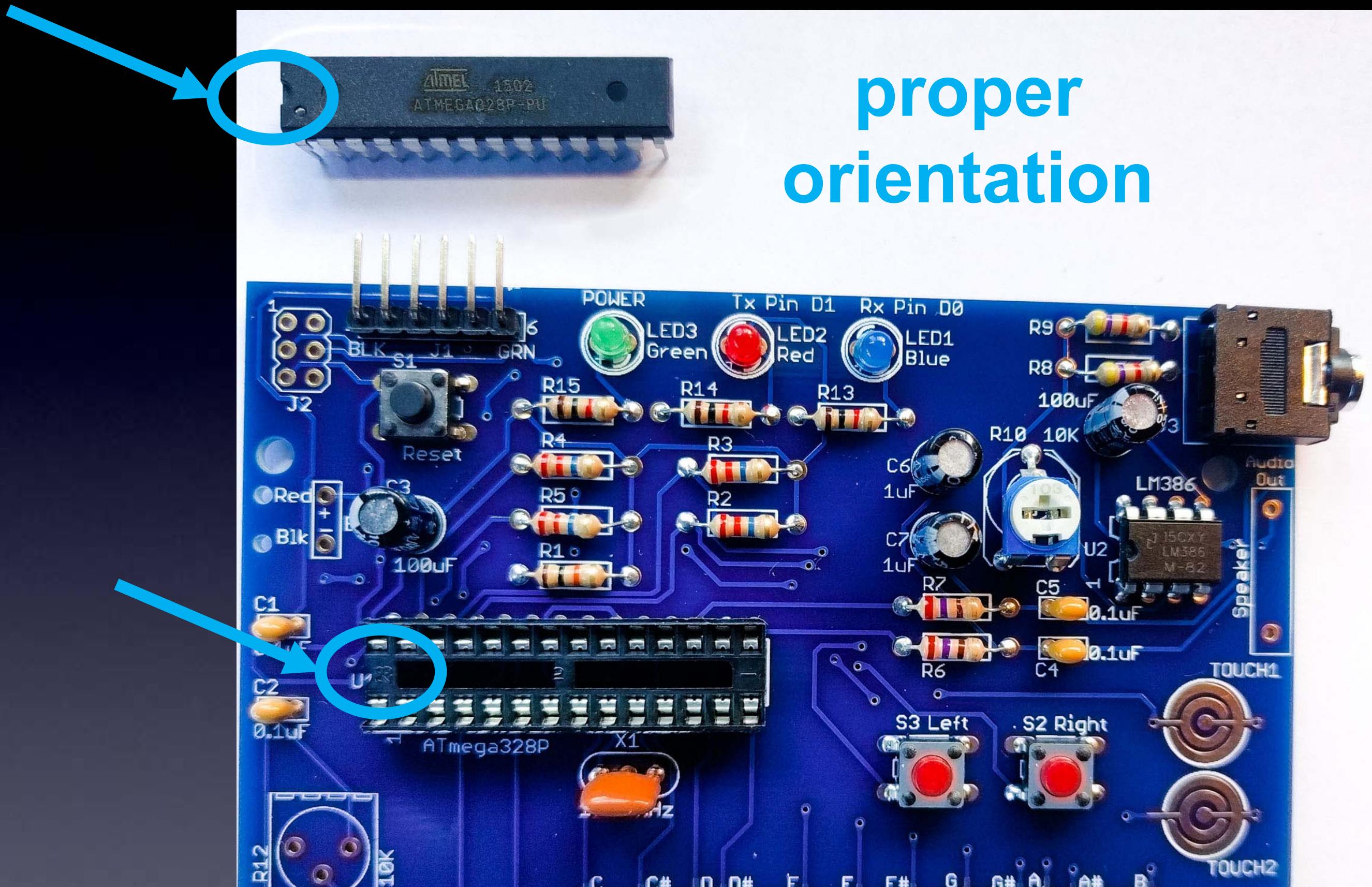


**We need the pins bent straight and parallel.
Use your work table to (gently) bend the leads.**



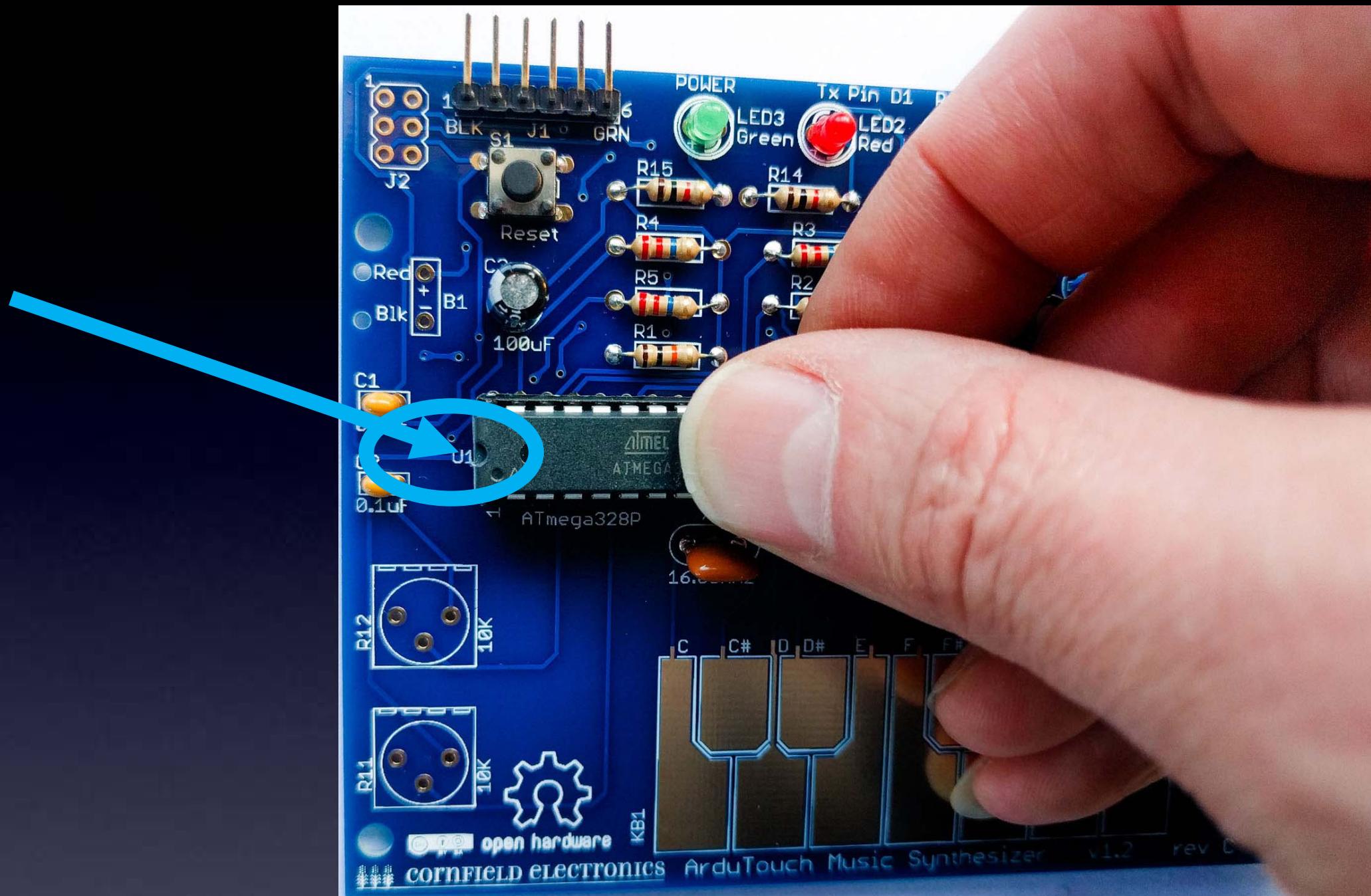
U1: microcontroller

These pins must be straight and parallel



proper orientation

U1: microcontroller



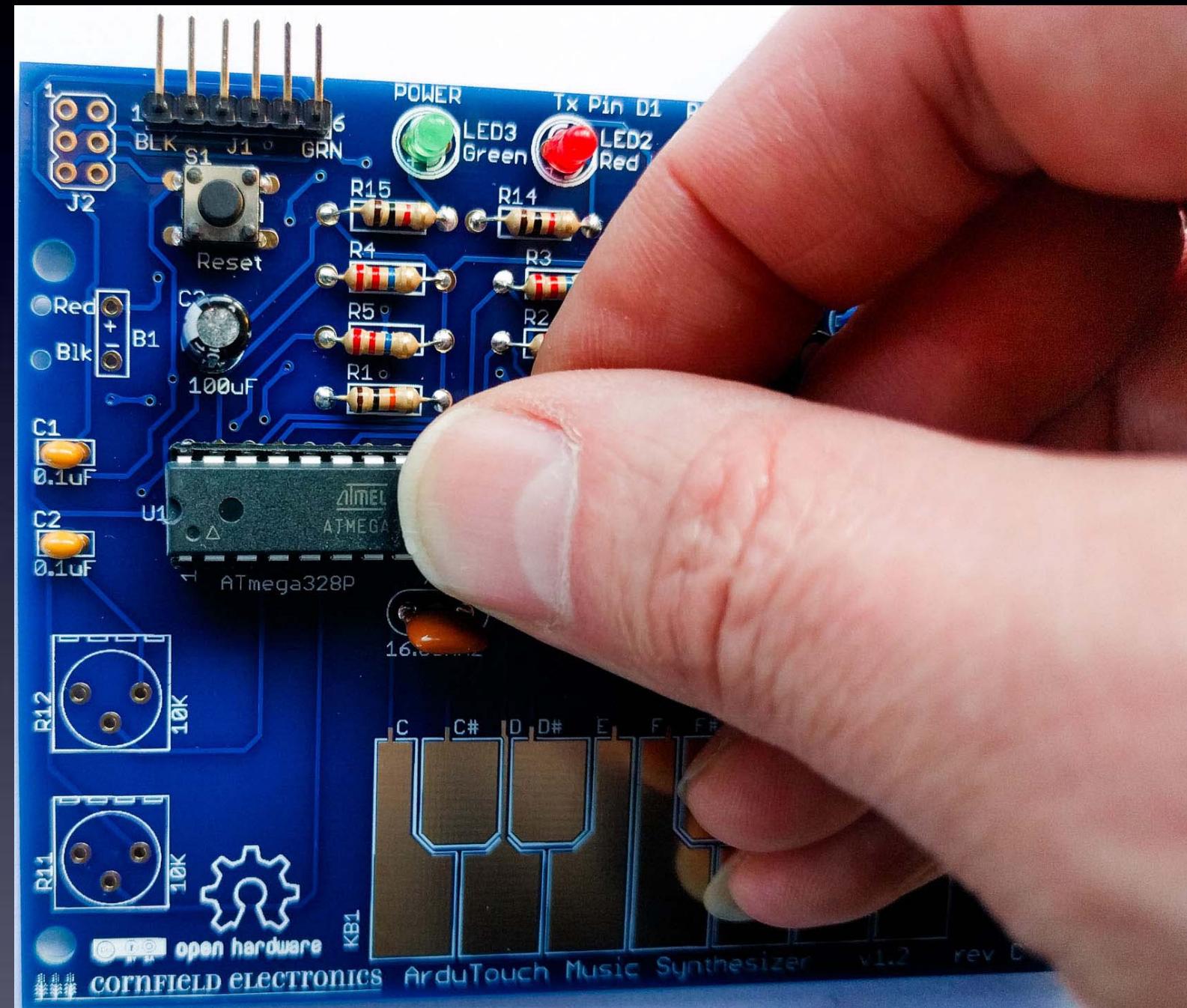
U1: microcontroller

make sure each pins rests in its hole in the socket
→ with the proper orientation

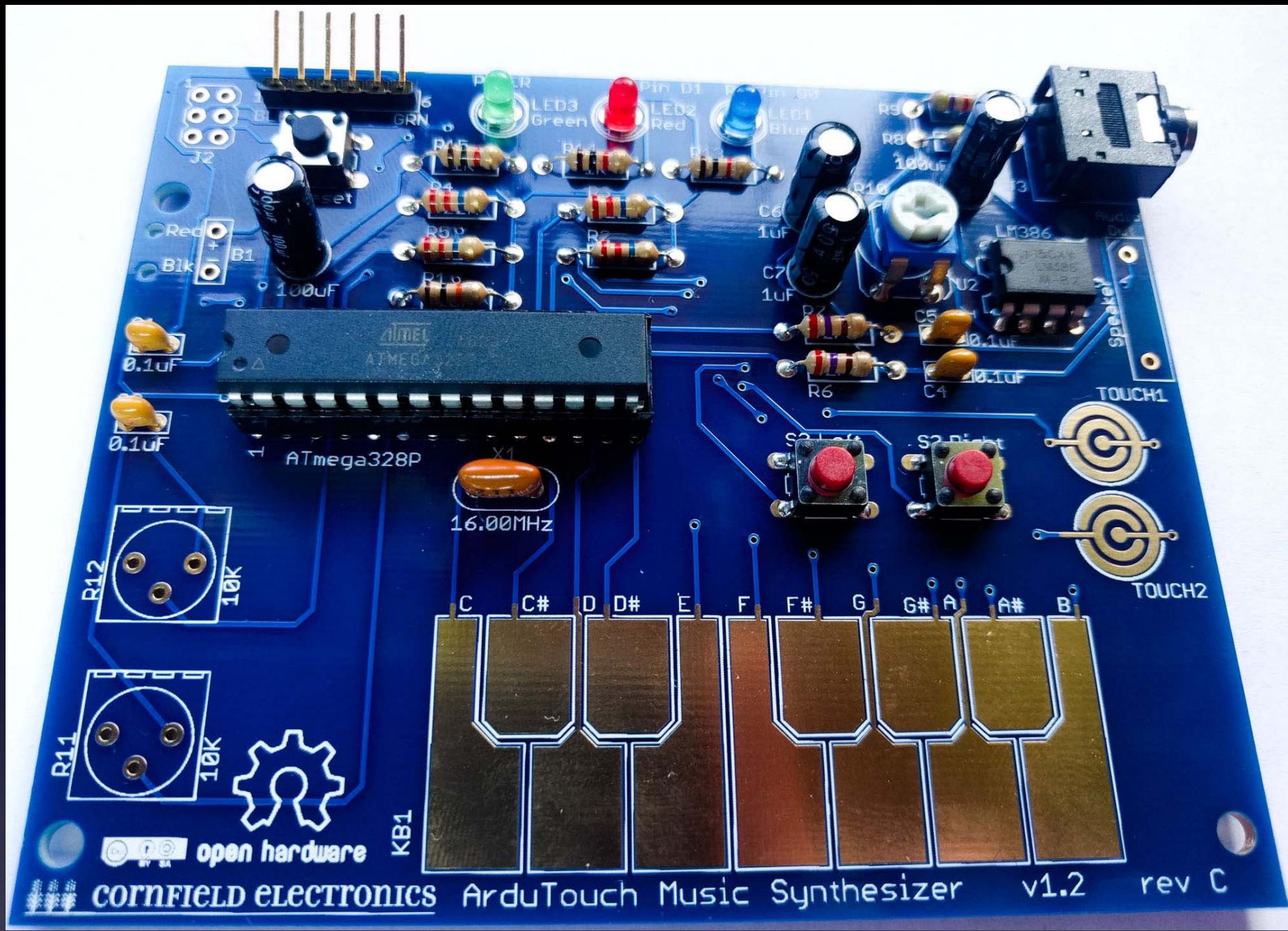
Use two thumbs to push microcontroller into the socket

**Make sure all 28 pins
are in place,
and push it into its socket.**

**(This is actually way easier
with 2 thumbs.)**



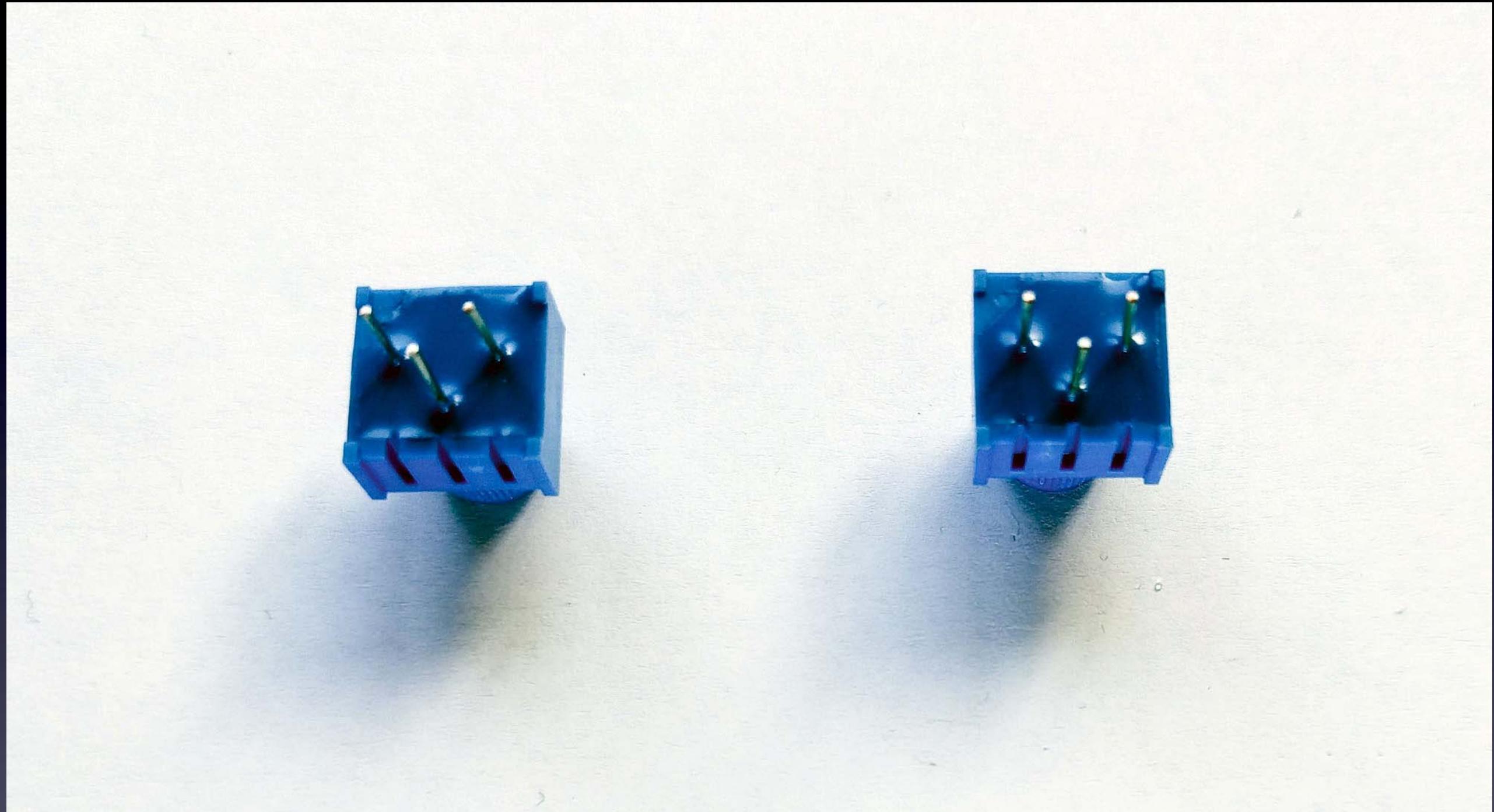
U1: microcontroller



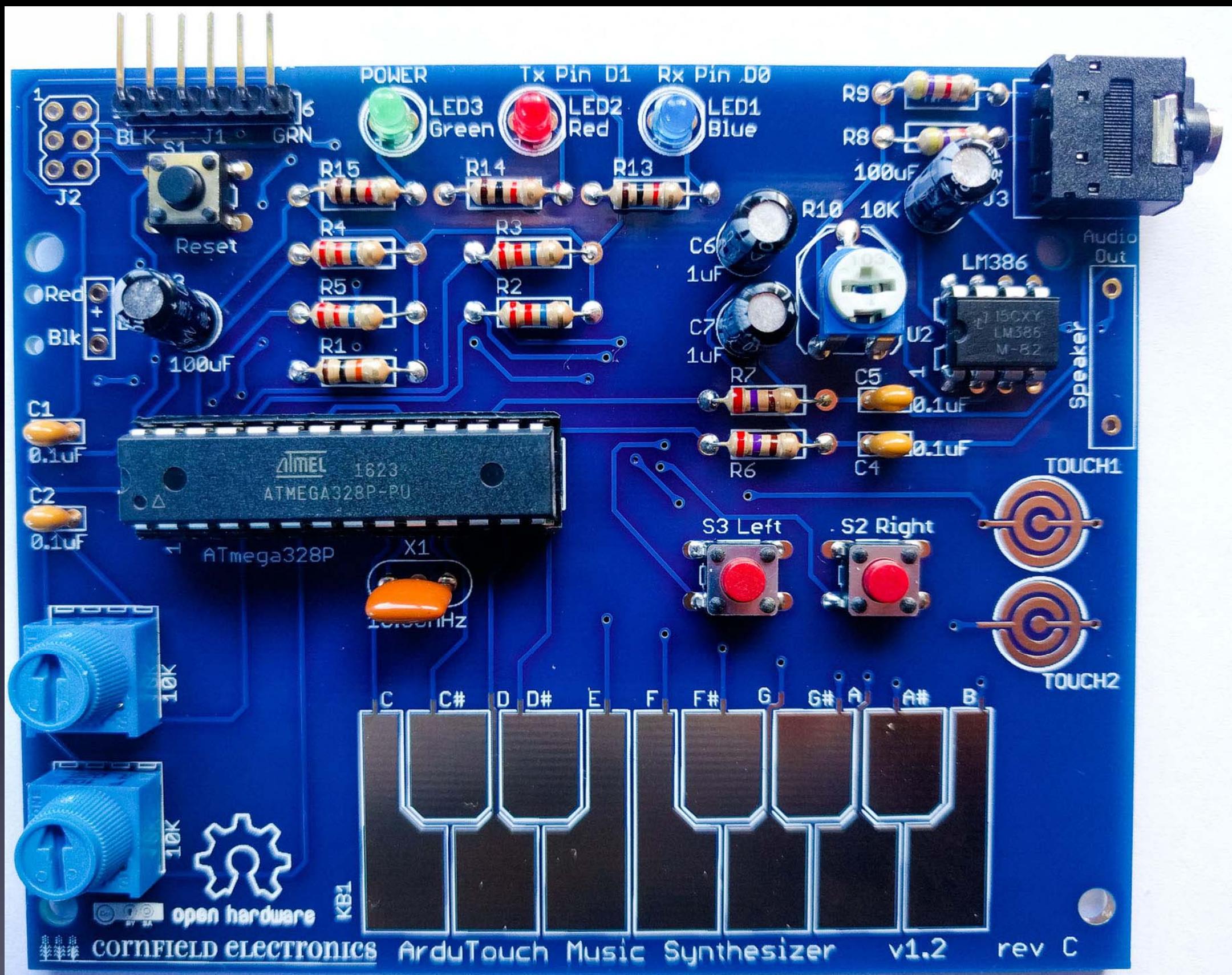
U1: microcontroller

Inspect all pins, and be sure each went into its hole in the socket – not bent.

If any pins are bent, (gently) pry out chip, straighten pins, and insert again.



R11 & R12: potentiometers

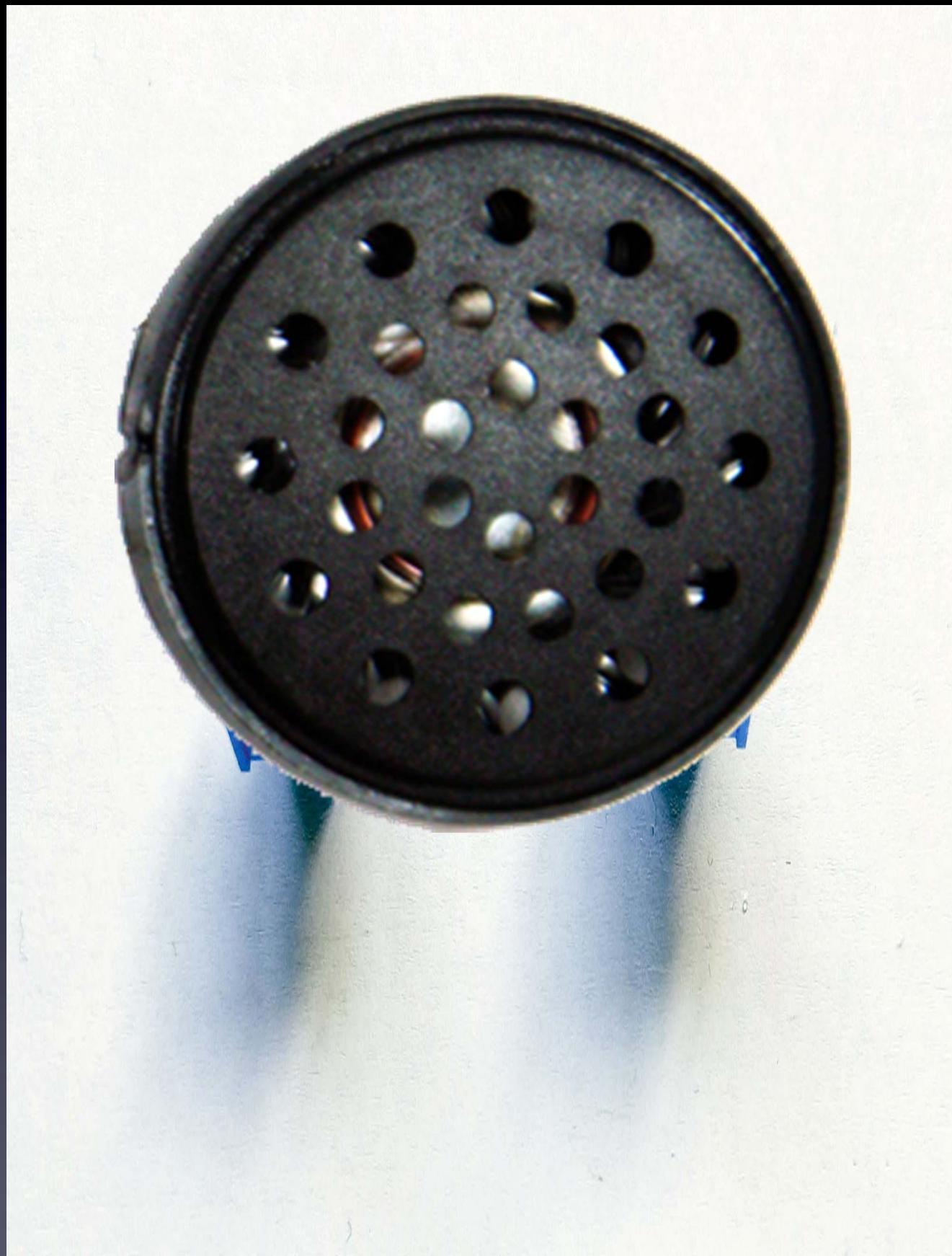


R11 & R12: potentiometers



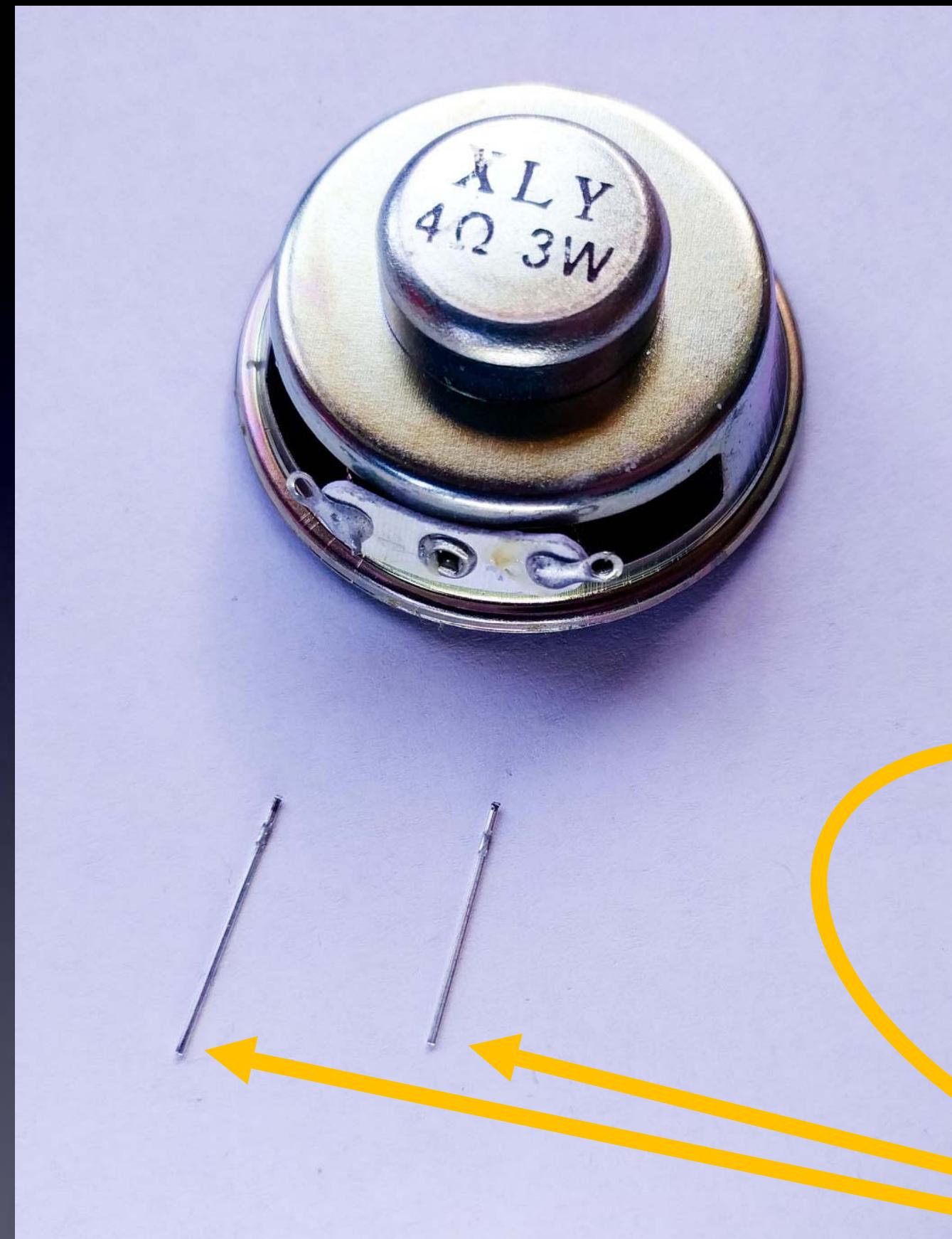
Speaker

**Some kits have a
speaker that looks
like this**



Speaker

We'll add leads
to the speaker



Saved
leads

from the LEDs

Speaker

**Tin one side
of each lead**

(i.e., cover with
thin film of melted solder)

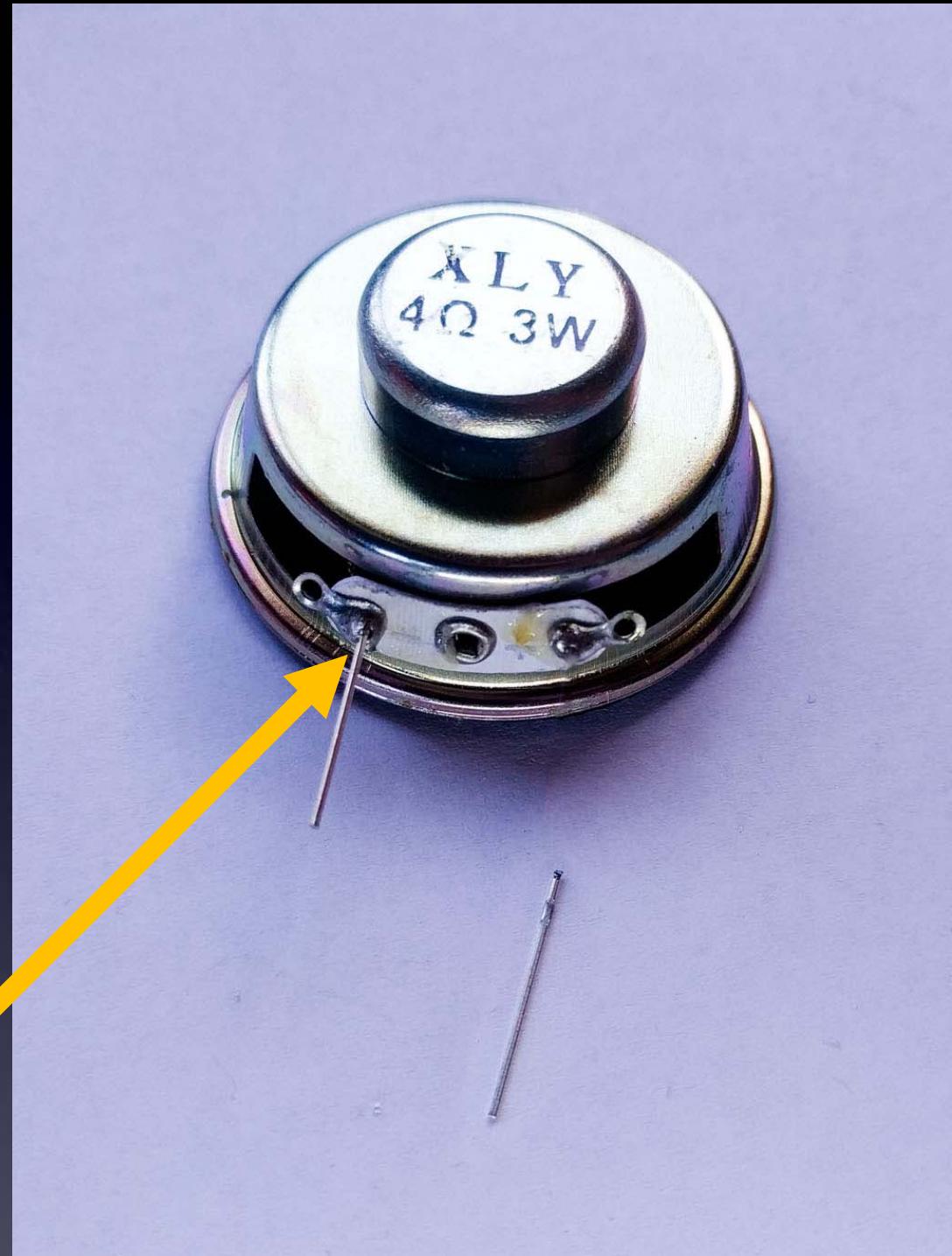


Speaker

**Solder one lead
to speaker**

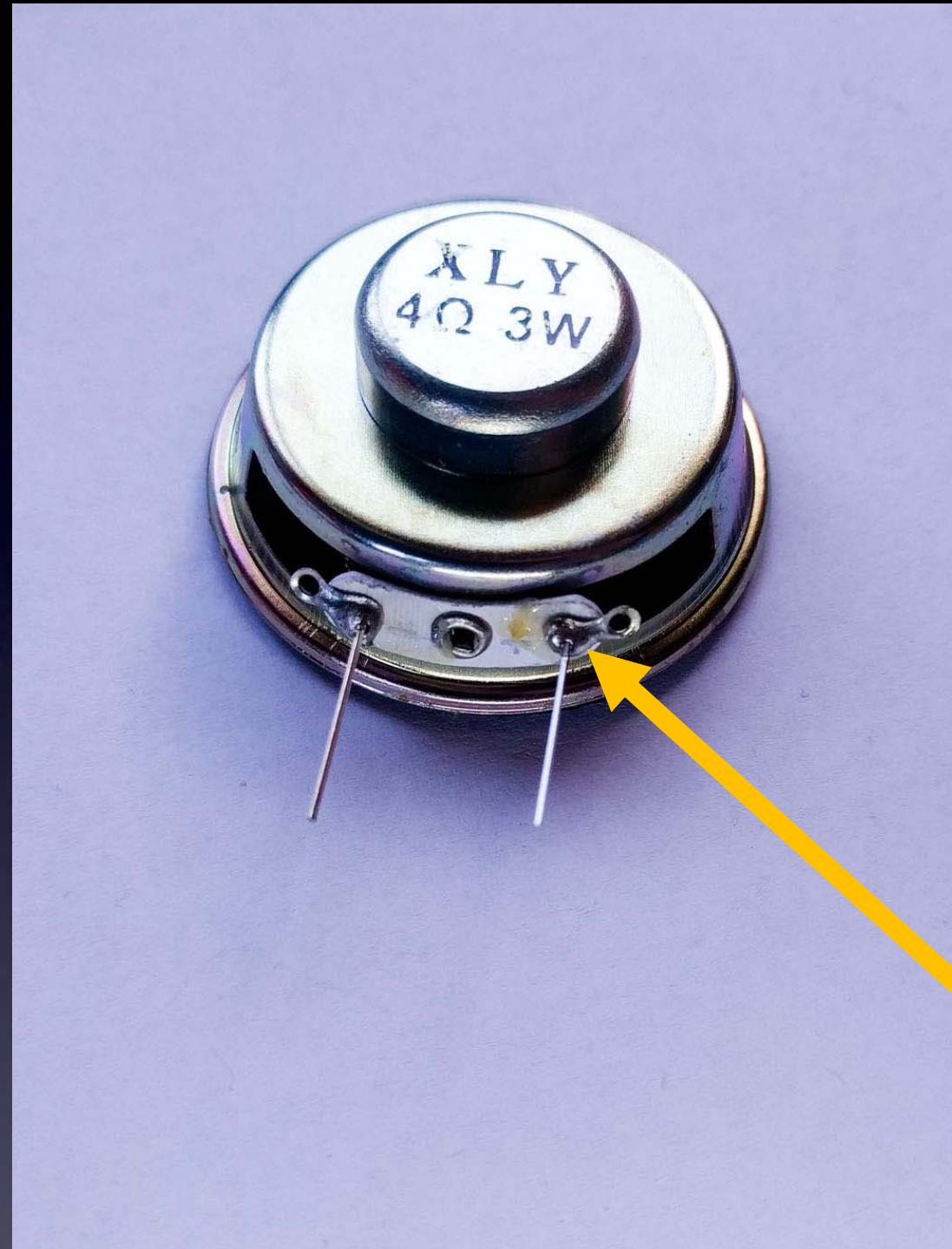


**Notice the
correct place
to solder the wire**



Speaker

**Solder next lead
to speaker**

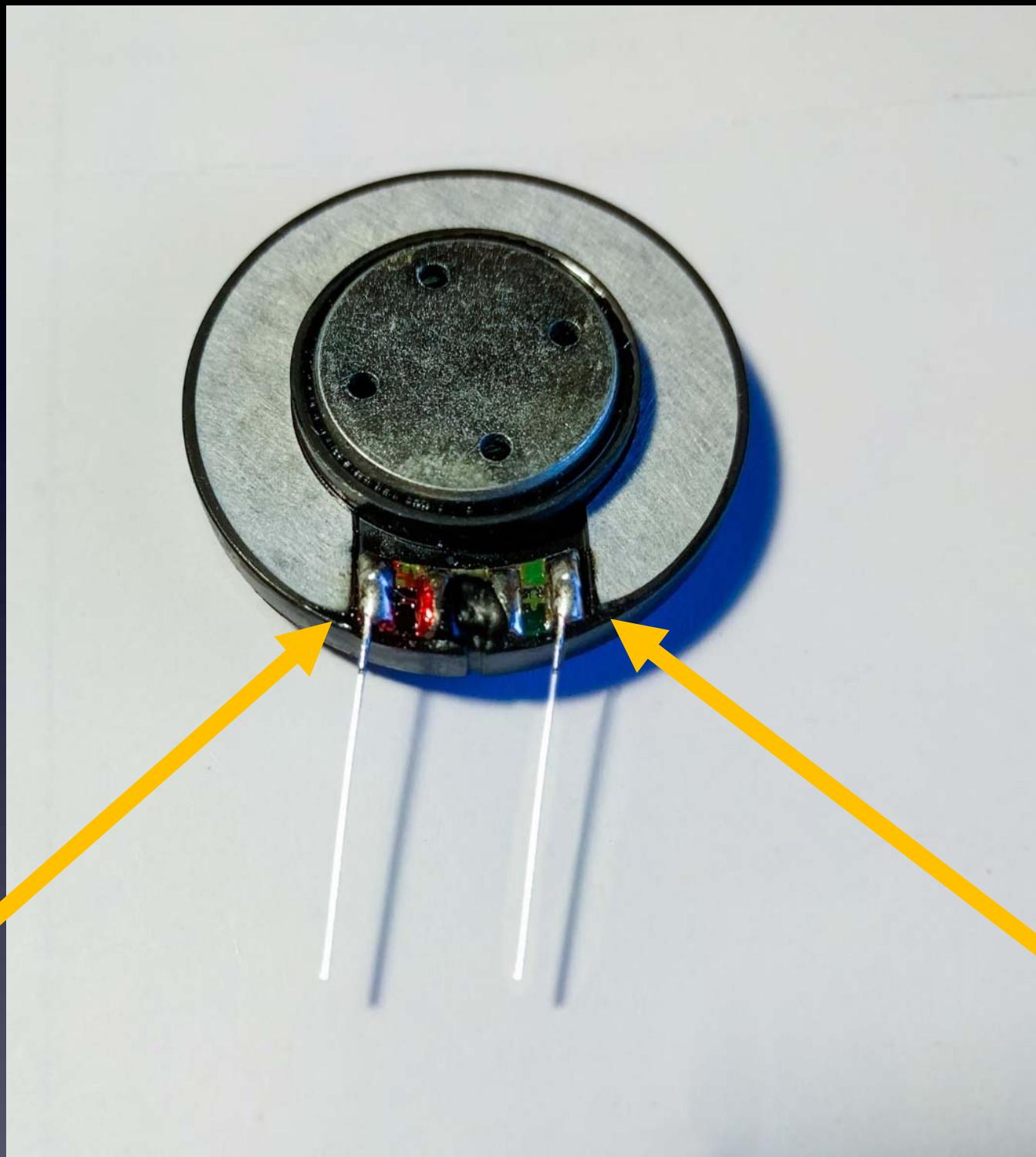


Speaker

**Notice the
correct place
to solder the wire**

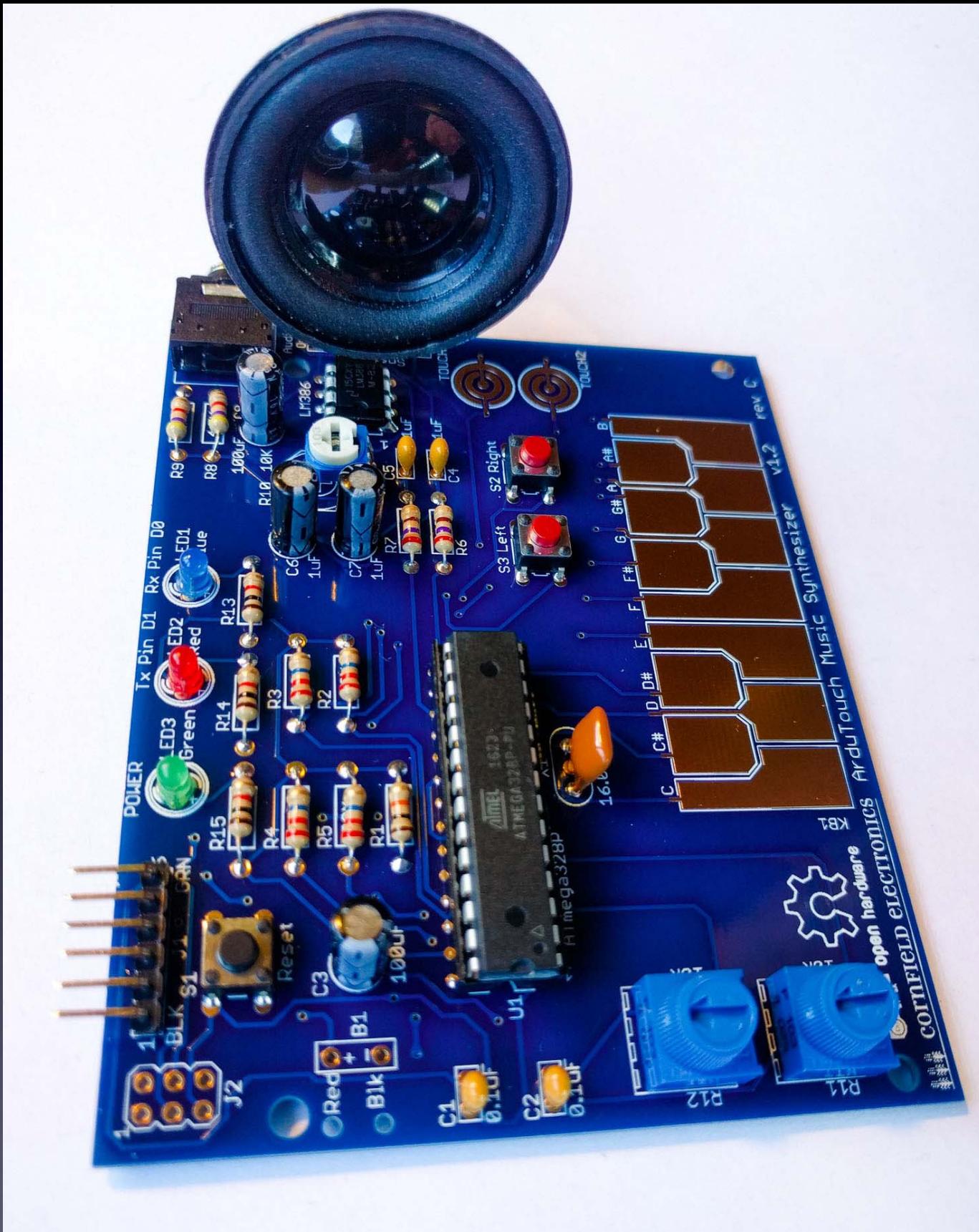
Some kits have a speaker that looks like this

Notice the correct place to solder the wires



Speaker

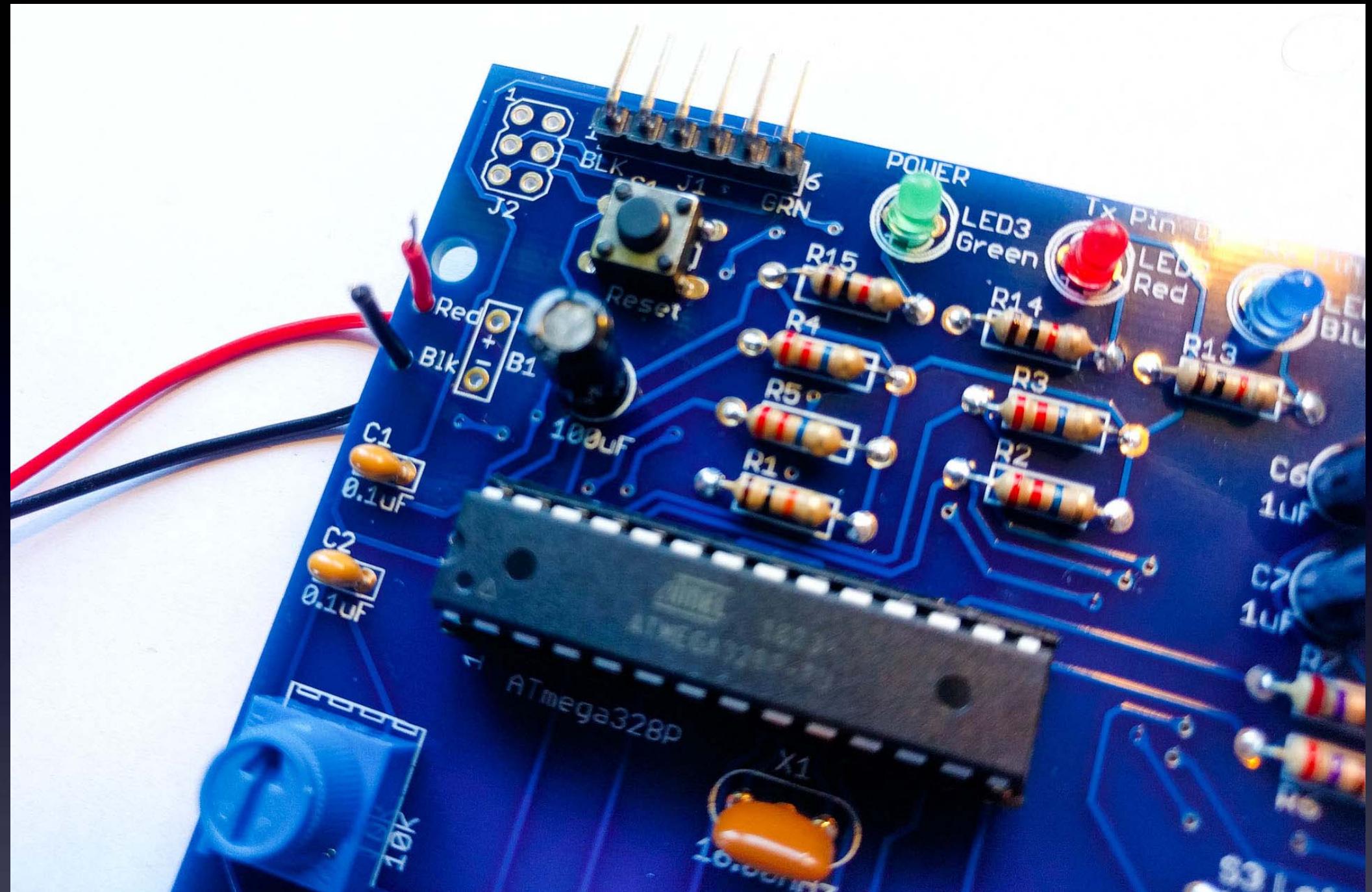
**Insert
speaker into board
and solder
both leads to board.**



Speaker

Note: Some battery pack wires have thicker red and black plastic coatings.

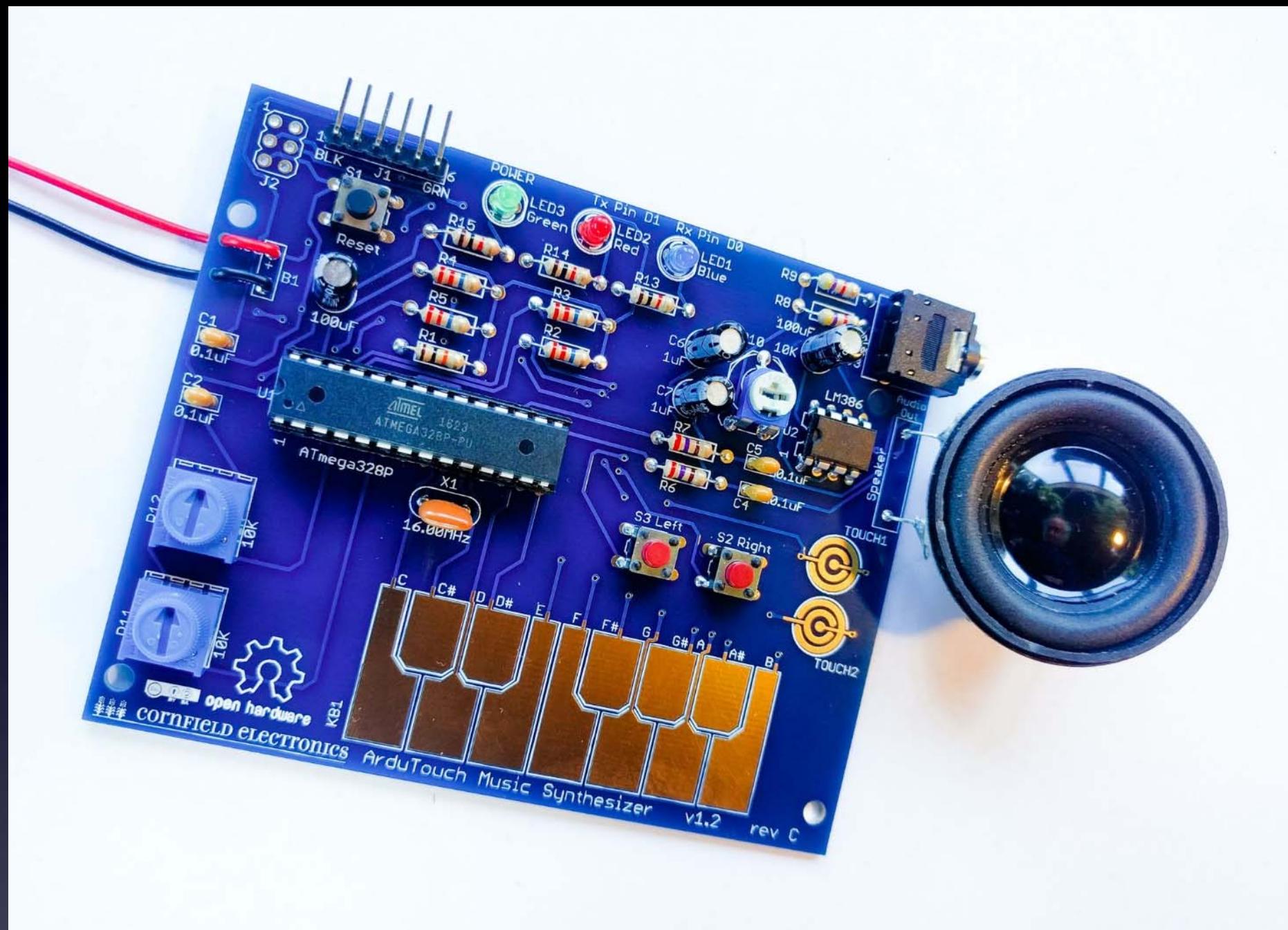
*If so,
you can widen the these
two holes by gently
rotating a scissors or small
knife or small Phillips
screwdriver on the top
and bottom of these two
holes.*



Push battery pack leads through holes.

Make sure Red and Black go through their correct holes!

Battery pack

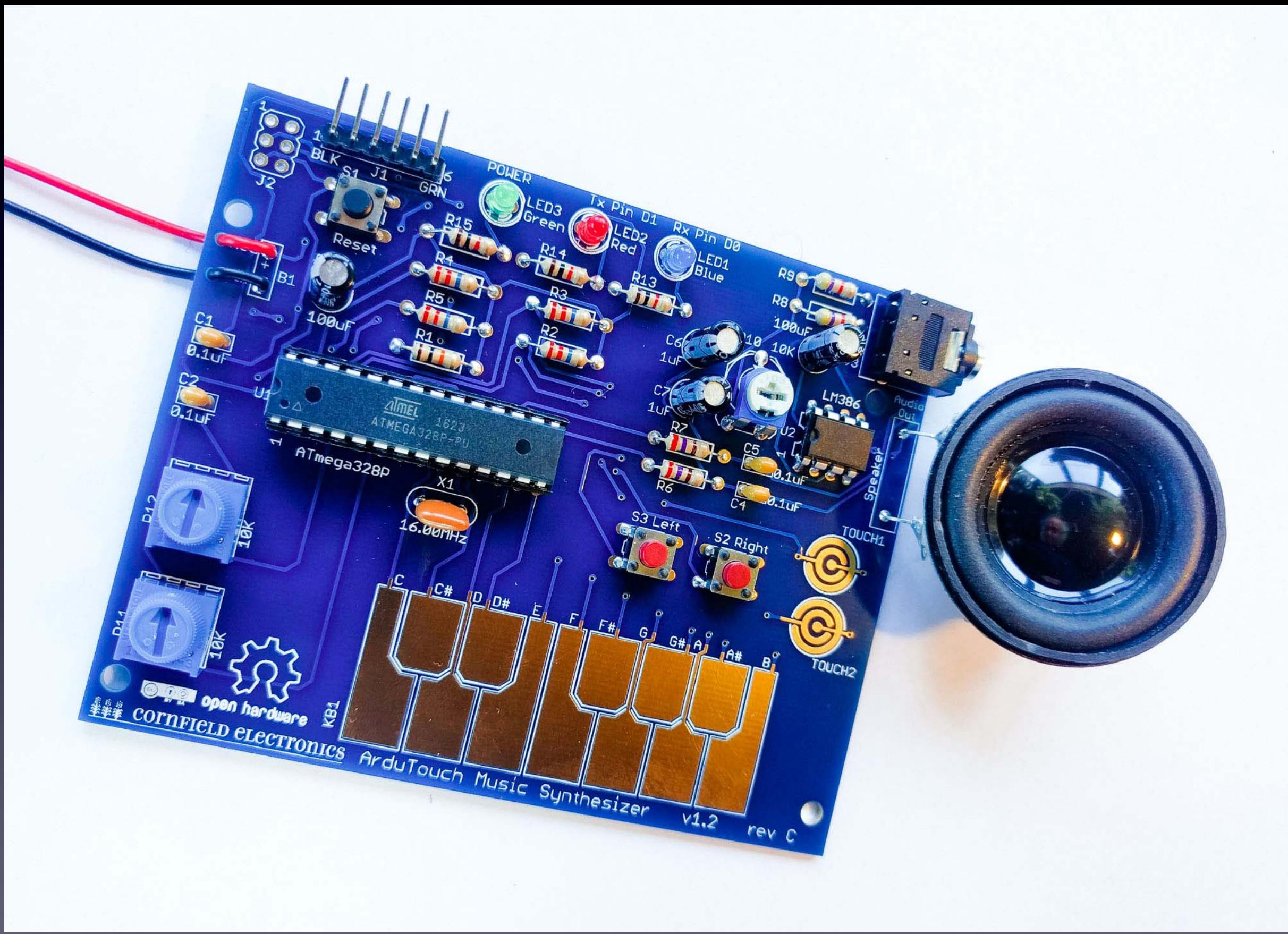


Loop one lead into its pad,
and solder.

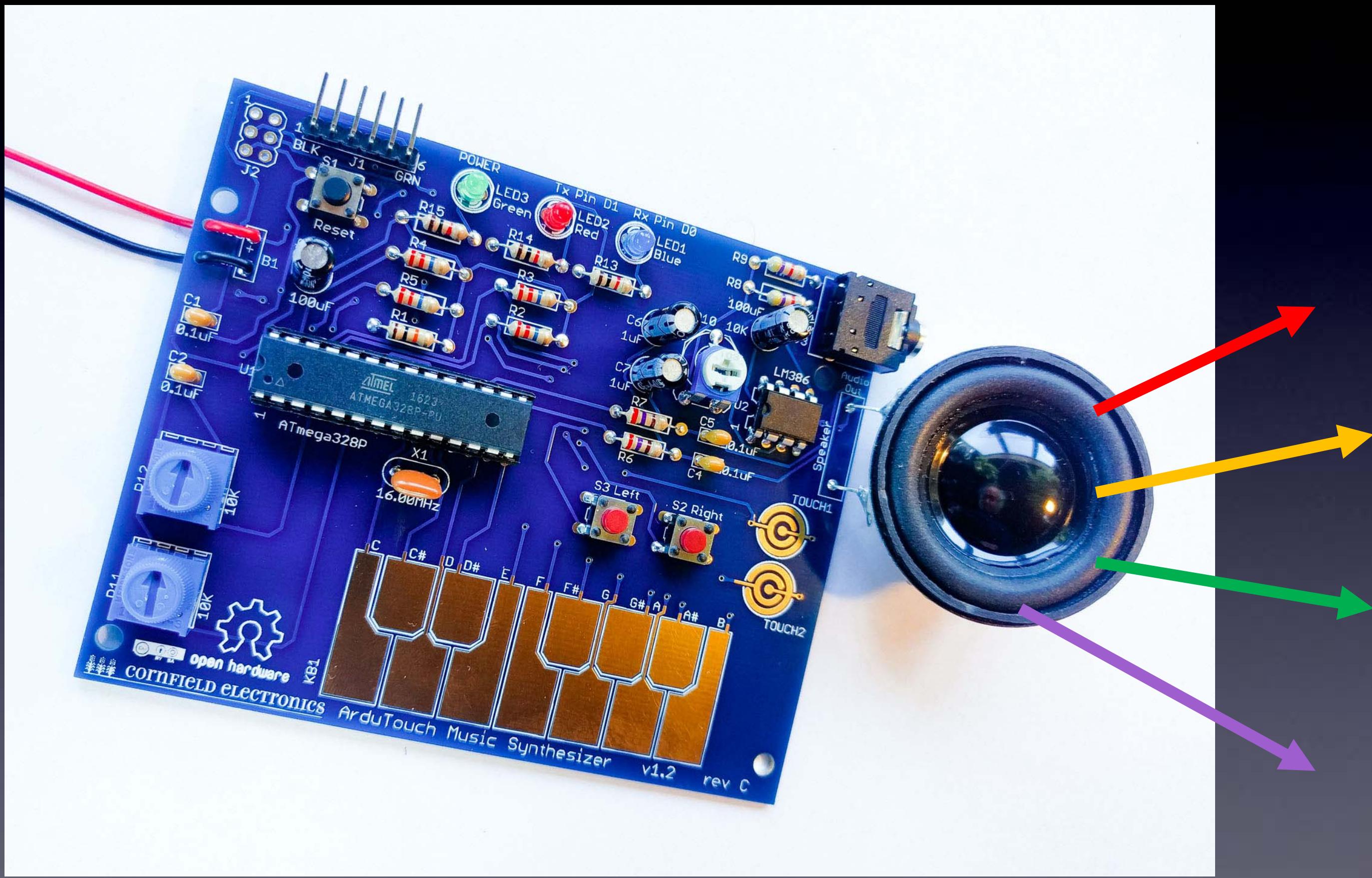
Then loop the other lead into its pad,
and solder.

Battery pack

Done!



Let's make noise!



Please Remember:

to

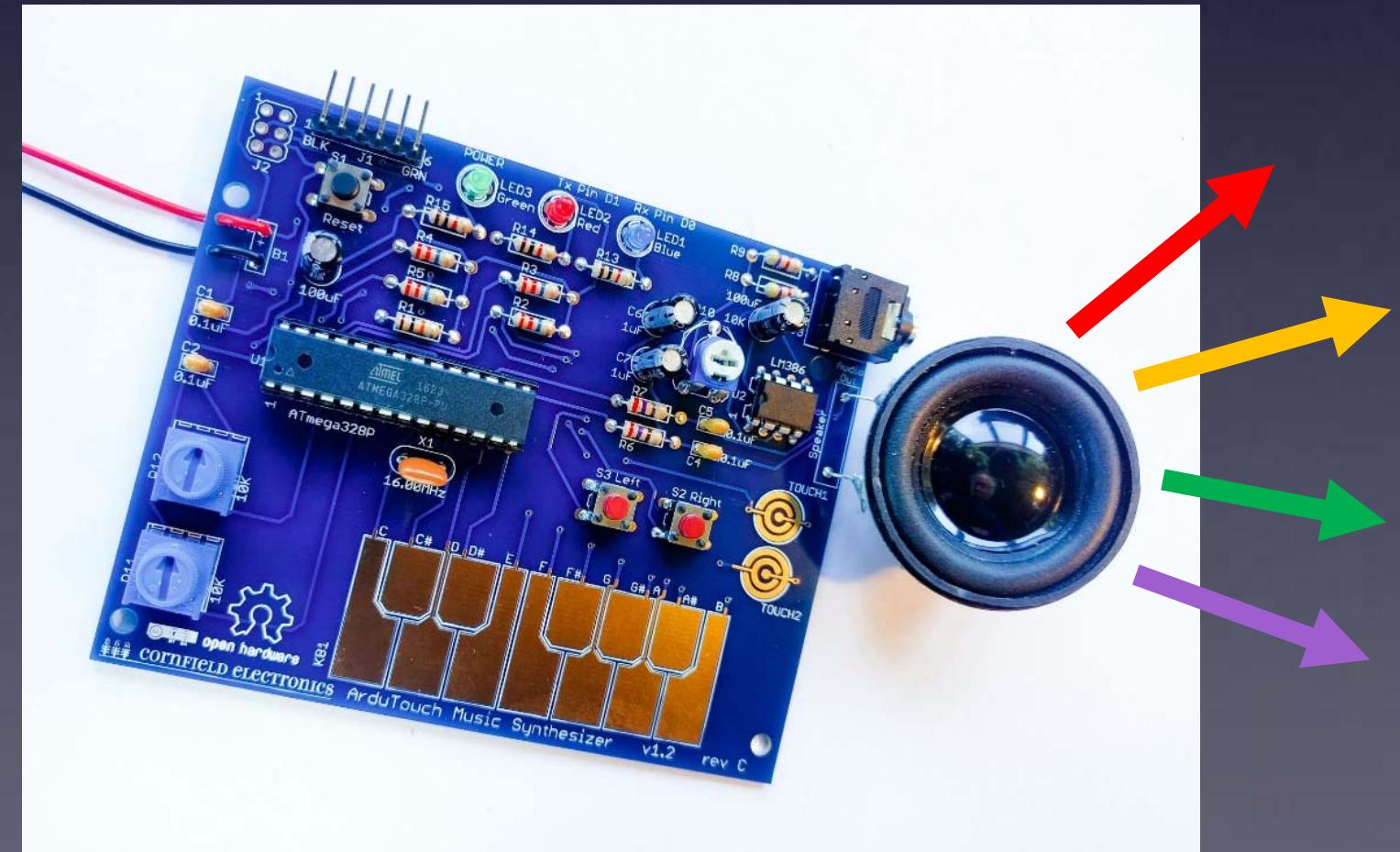
Wash your hands
after soldering

Let's make noise!

Your ArduTouch comes pre-programmed with a really cool synthesizer, called “Thick”.

“Thick” plays 4 sawtooth waves at once.

- the left and right buttons change octaves
- long press the left and right buttons to change sounds
- the Bottom knob controls the glide rate
- the Top knob controls how each of the 4 notes glide separately
- Try playing with these and see!

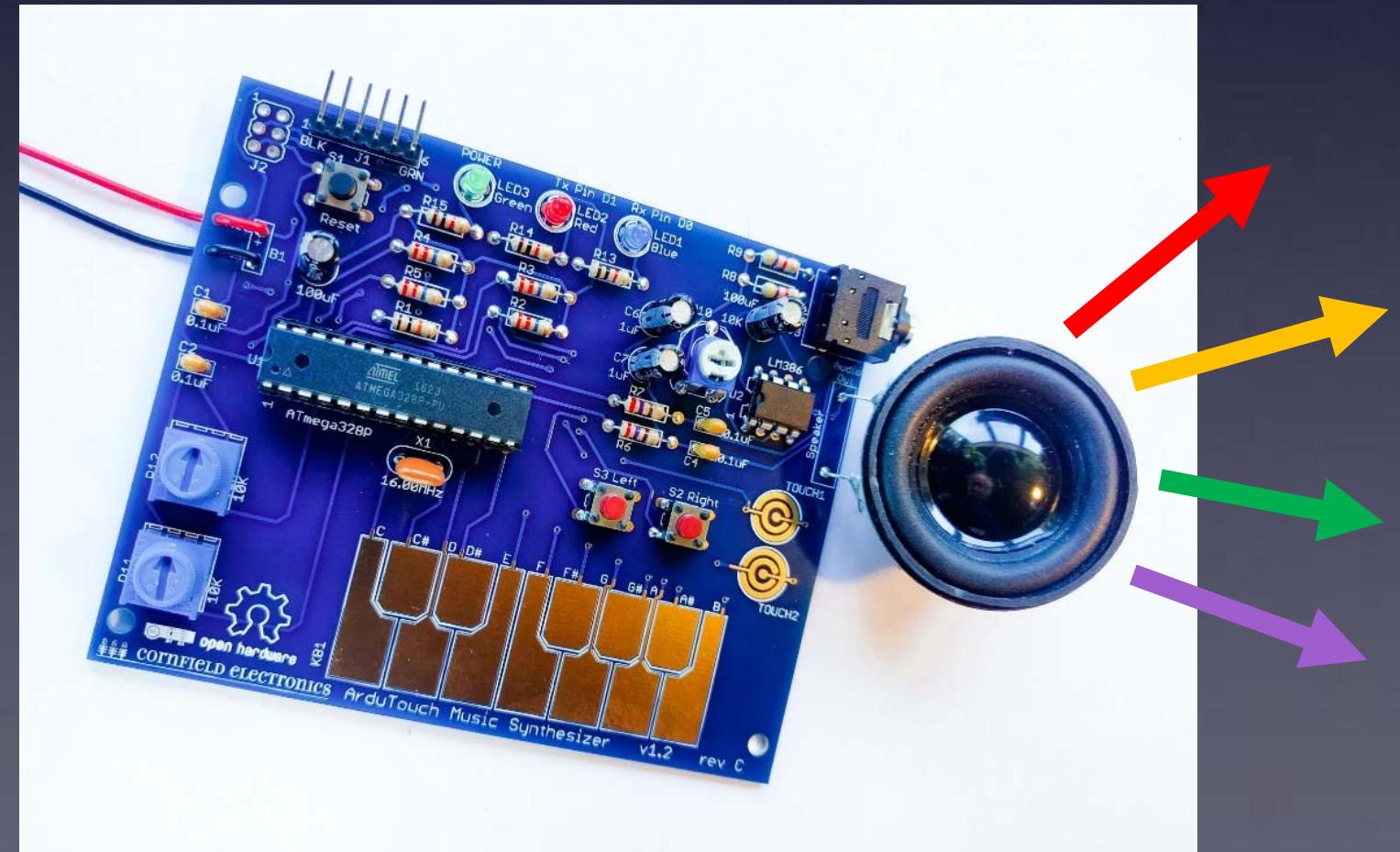


Let's make noise!

Your ArduTouch comes pre-programmed with a really cool synthesizer, called “Thick”.

If you are happy playing with “Thick”
then no need to re-program
your ArduTouch.

But if you want to
program other synths into your
ArduTouch,
the next pages show you how...

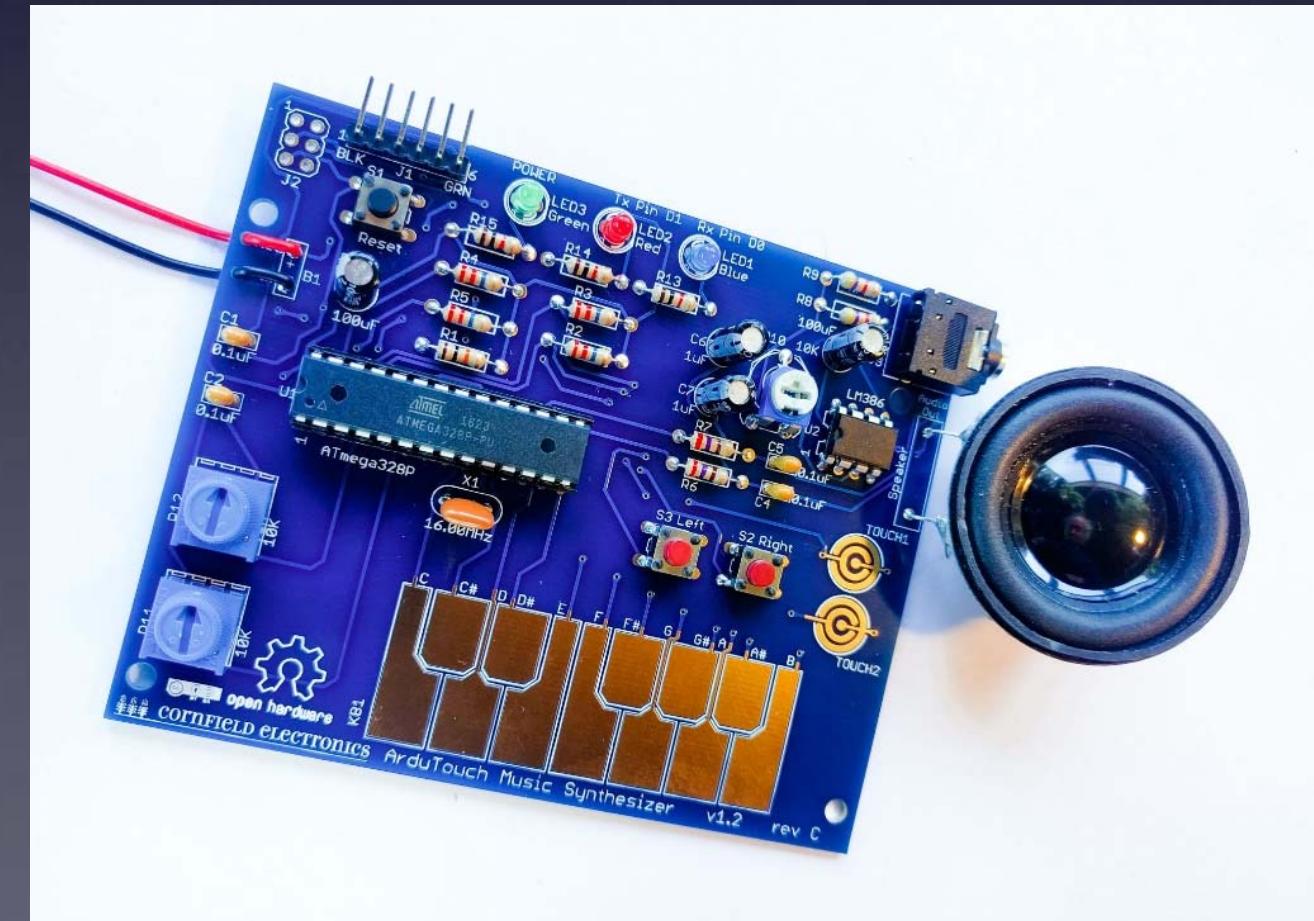


Re-programming the ArduTouch

We have written several way cool synthesizers for ArduTouch!
Each is unique, and each way different than the others.

To program in a new synth in your ArduTouch, you will need:

- the Arduino software <<http://arduino.cc>>
- a USB-Serial adapter cable (such as an FTDI, or equivalent)
a nice one is available at
<<https://cornfieldelectronics.com/cfe/products/buy.php?productId=usbcable>>
- a synth sketch and the ArduTouch Arduino library
<[http://cornfieldelectronics.com/cfe/projects.php#ardutouch](https://cornfieldelectronics.com/cfe/projects.php#ardutouch)>



Arduino

**Arduino is a very powerful tool!
But it is very easy to use.
It was designed for total beginners to use successfully.**

I won't give a complete tutorial here – just some basics.
For more info, there are many good Arduino tutorials online.
A good place to start is:
[<https://www.arduino.cc/en/Tutorial/HomePage>](https://www.arduino.cc/en/Tutorial/HomePage)



Intro to Arduino



Arduino For Total Newbies workshop

Day 3 Tuesday 29-December, 13:00 to 16:30

→ → **Right-click on this link, and open in a new window**
[Arduino For Total Newbies workshop room on Big Blue](#)

NOTE: You do NOT need to register to take this workshop.
Just show up before the start time at the Big Blue Button room, given above.

*Learn Arduino
using TV-B-Gone
as an example project
(no materials required)*



Arduino For Total Newbies workshops

Arduino

First:
Download and install the Arduino software
< <http://arduino.cc> >

Any version is OK

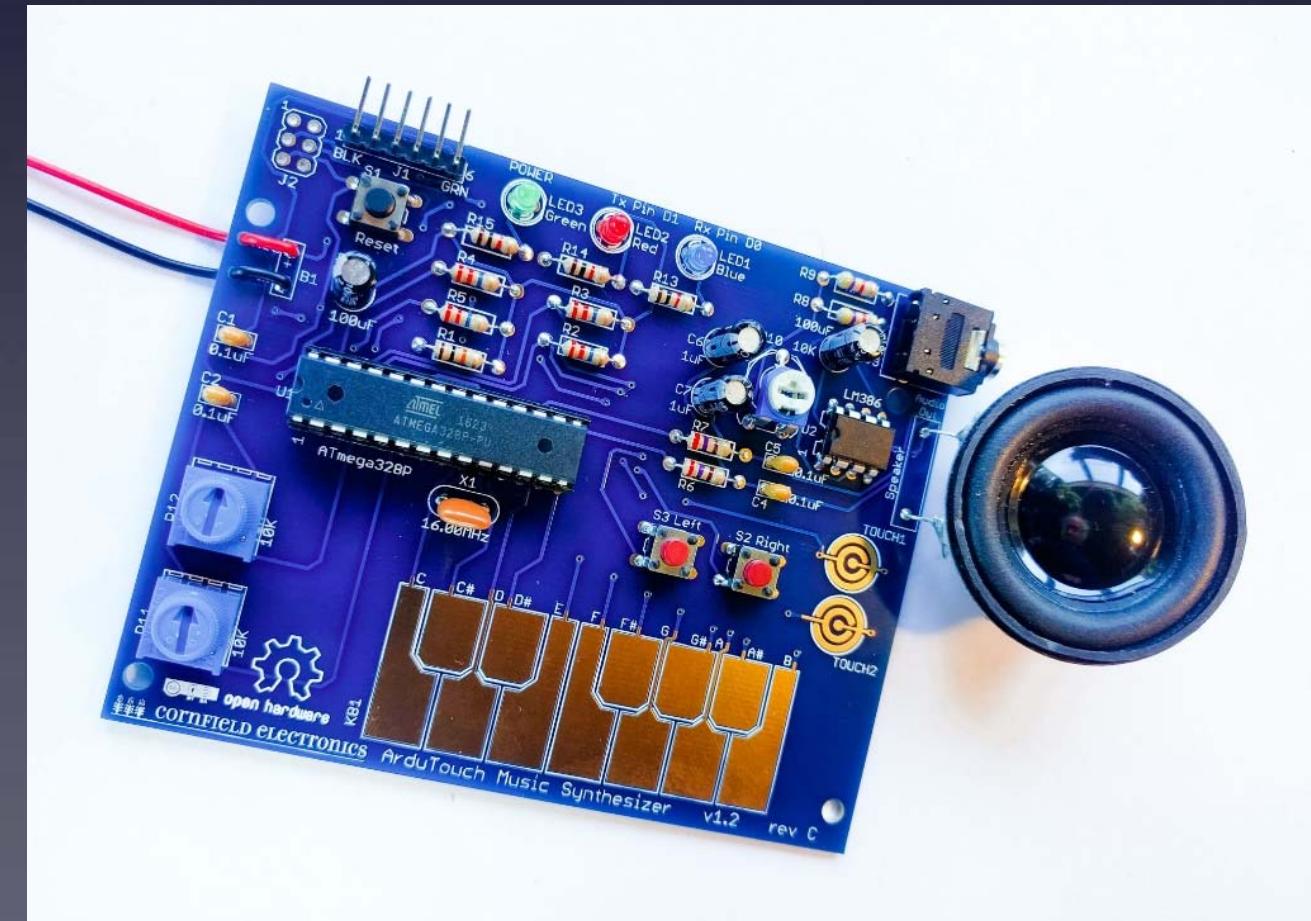


Re-programming the ArduTouch

Second:

Download and install the ArduTouch Arduino library
[<http://cornfieldelectronics.com/cfe/projects.php#ardutouch>](http://cornfieldelectronics.com/cfe/projects.php#ardutouch)

(details on this soon)



Re-programming the ArduTouch

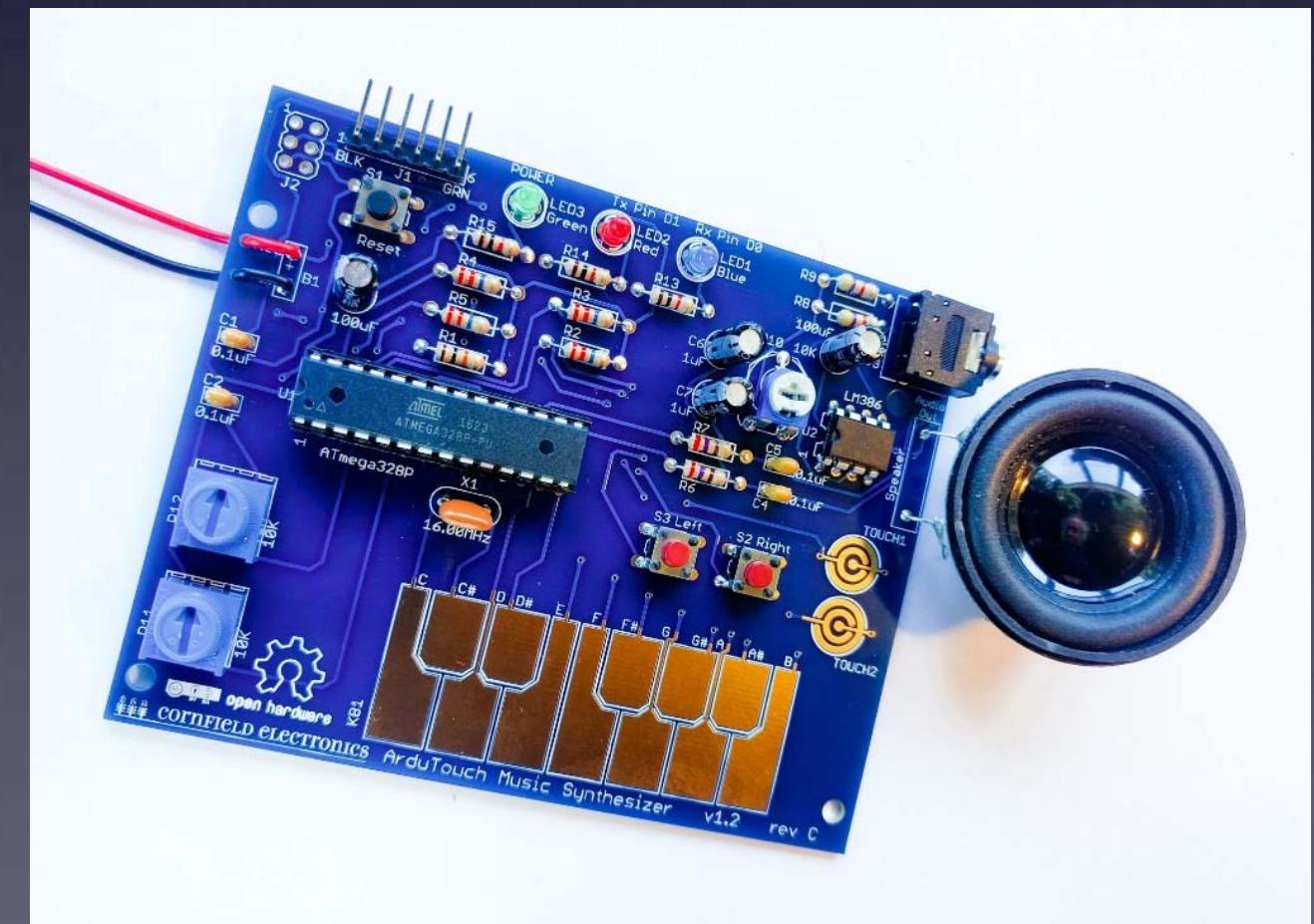
Third:

Download ArduTouch synth sketches

[<http://cornfieldelectronics.com/cfe/projects.php#ardutouch>](http://cornfieldelectronics.com/cfe/projects.php#ardutouch)

Store them on your computer anywhere you like.

(details on this soon)

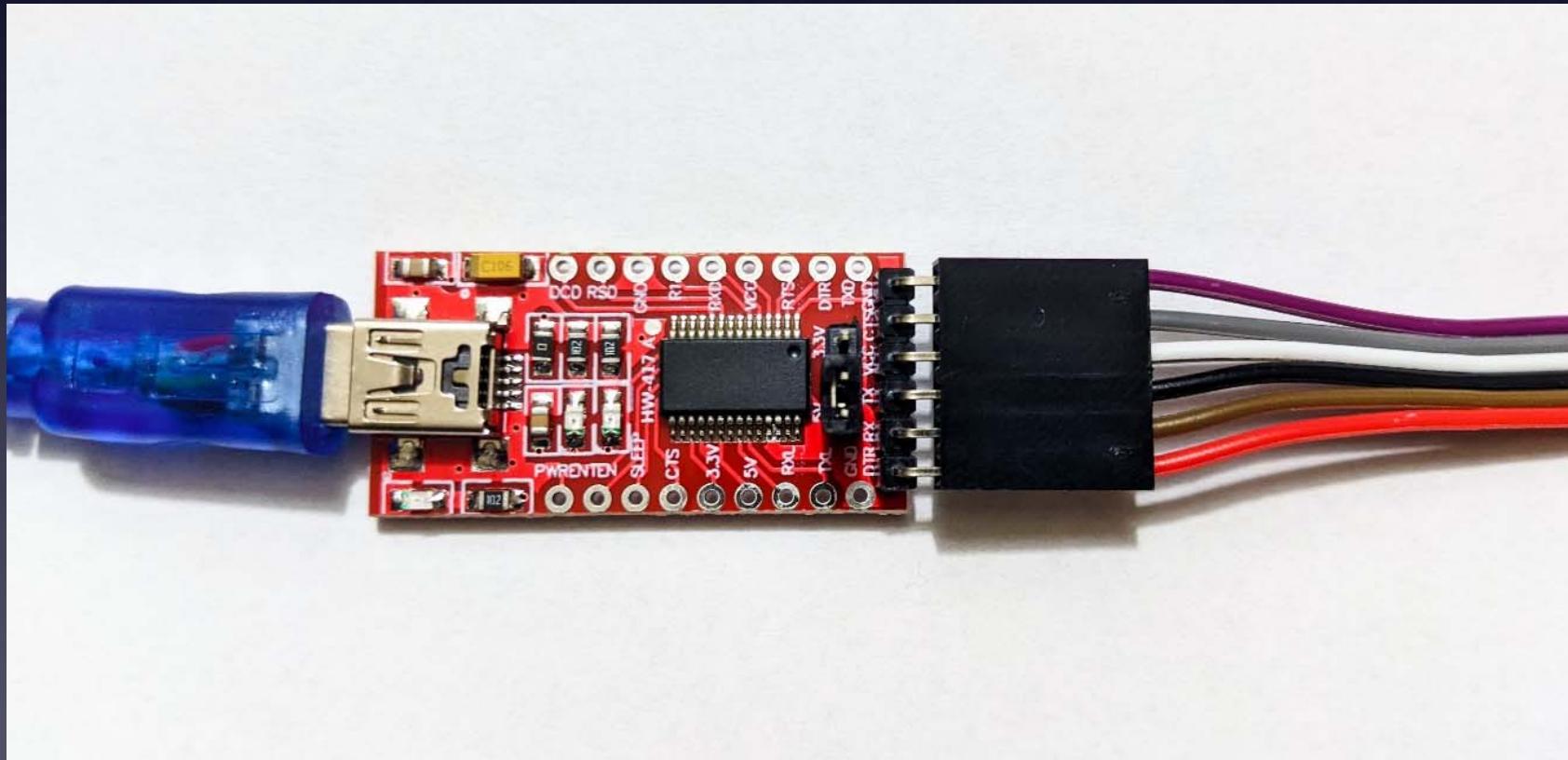


Connecting your ArduTouch to your computer

USB-Serial adapter cable

Ones available from Cornfield Electronics look like this:

<<https://cornfieldelectronics.com/cfe/products/buy.php?productId=usbcable>>

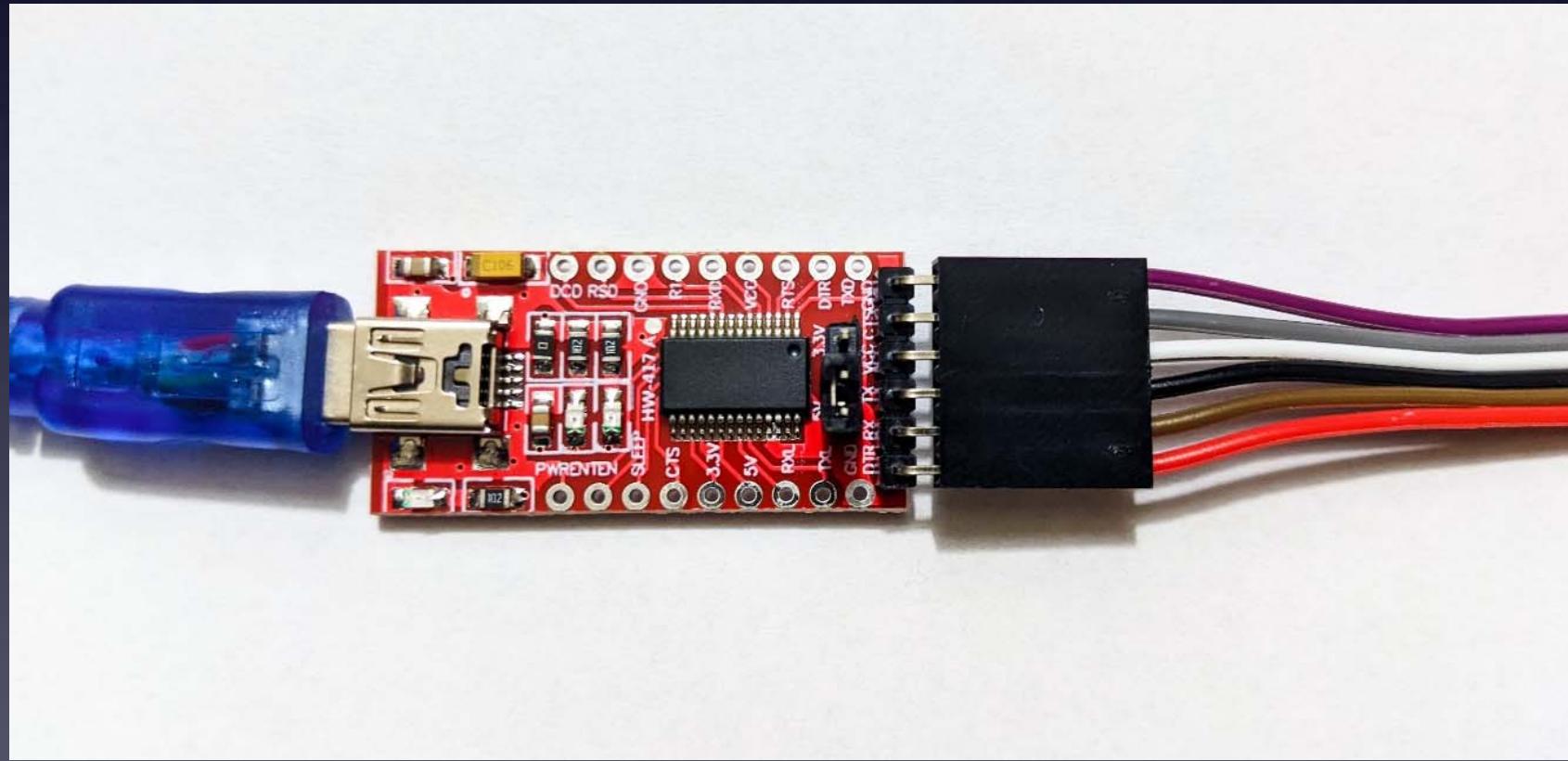


Connecting your ArduTouch to your computer

USB-Serial adapter cable

Ones available from Cornfield Electronics look like this:

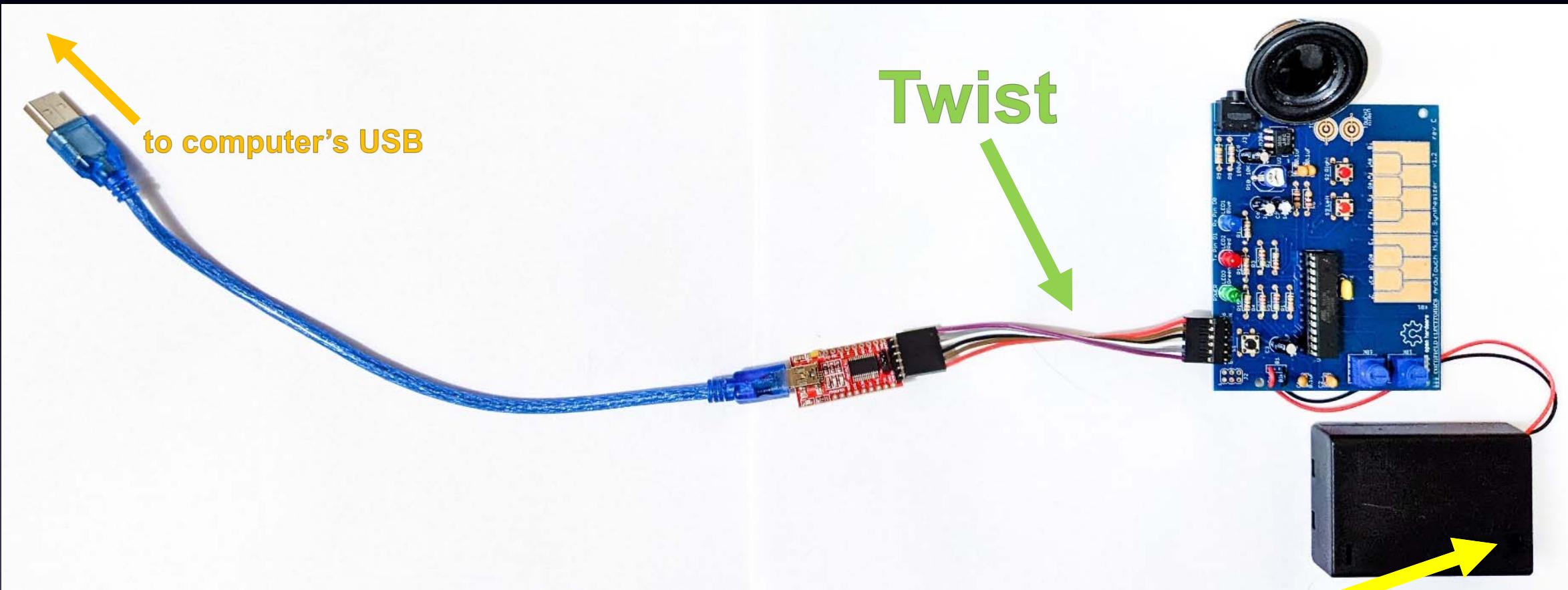
<<https://cornfieldelectronics.com/cfe/products/buy.php?productId=usbcable>>



**You may need to download and install a driver
for your Operating System (Windows, MacOS, or Linux):**

<<https://ftdichip.com/drivers/vcp-drivers/>>

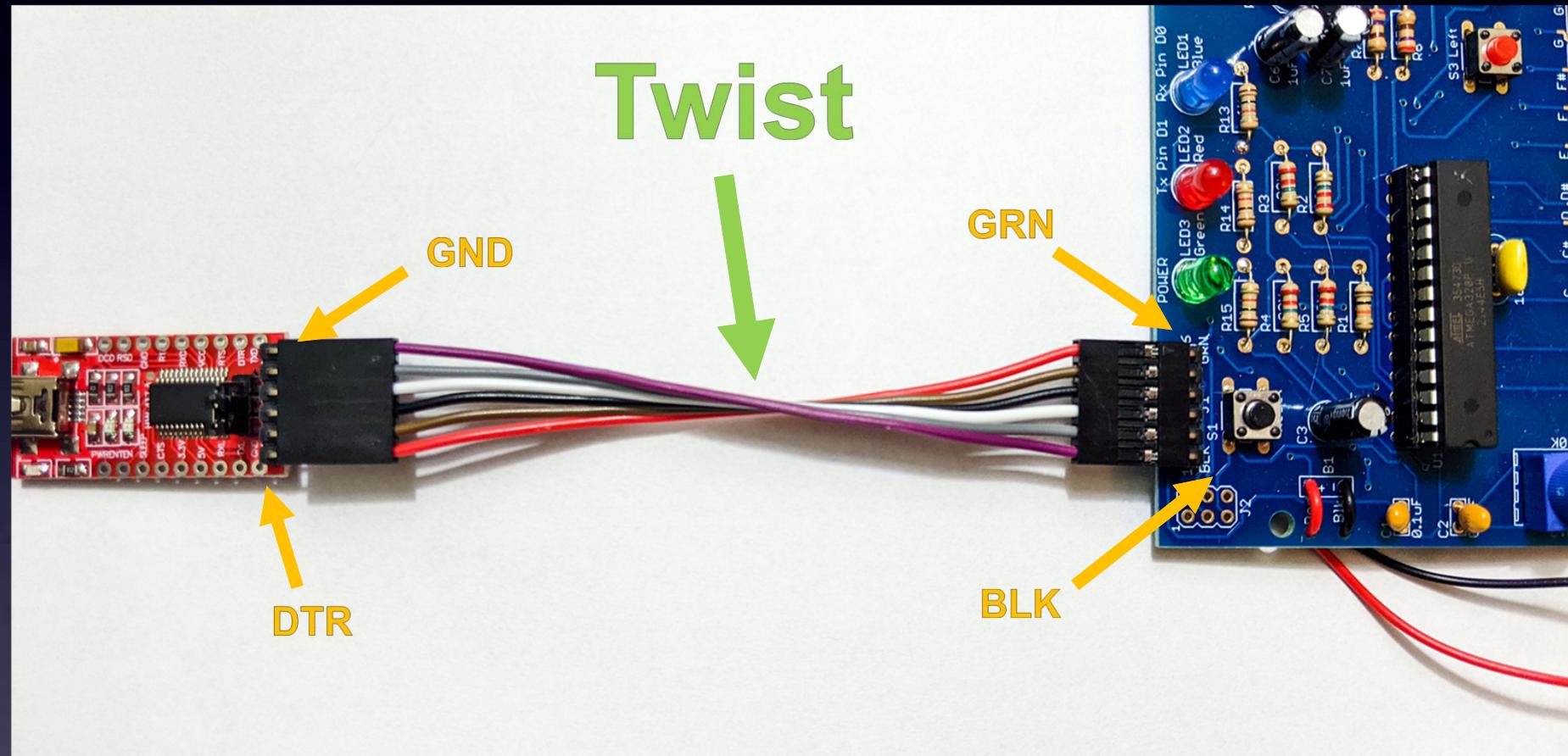
Connecting your ArduTouch to your computer



IMPORTANT:
Make sure the
battery pack on your
ArduTouch
is OFF

Connecting your ArduTouch to your computer

This shows a few more details:

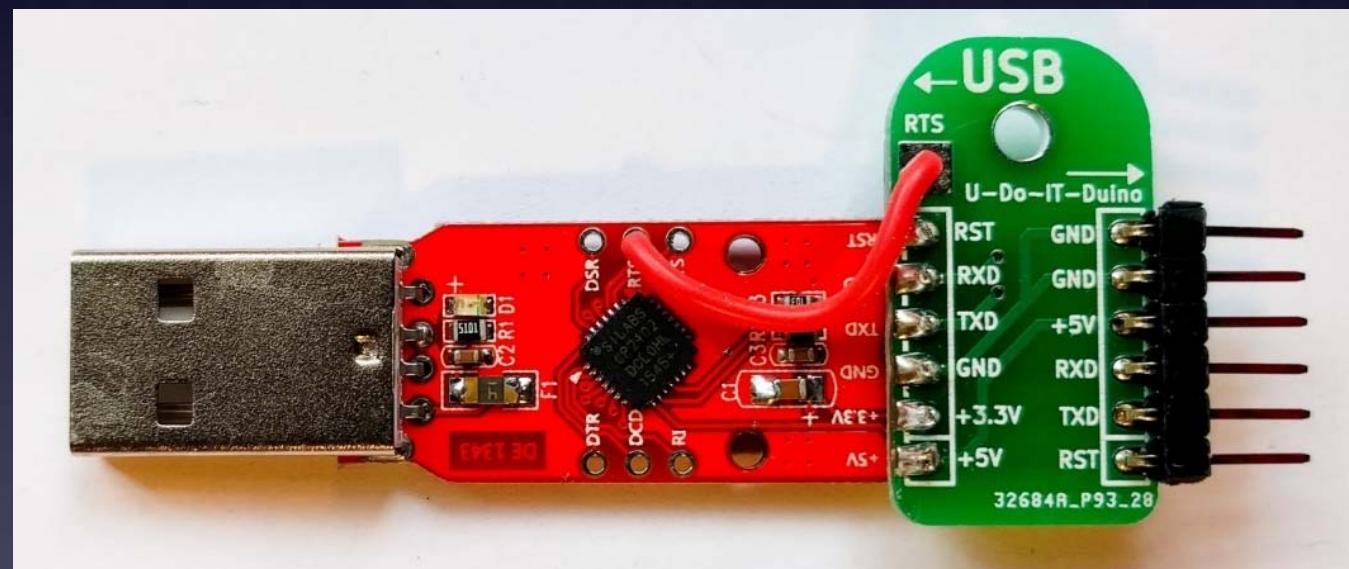


IMPORTANT:
Make sure the
battery pack on
your ArduTouch
is OFF

Connecting your ArduTouch to your computer

USB-Serial adapter cable

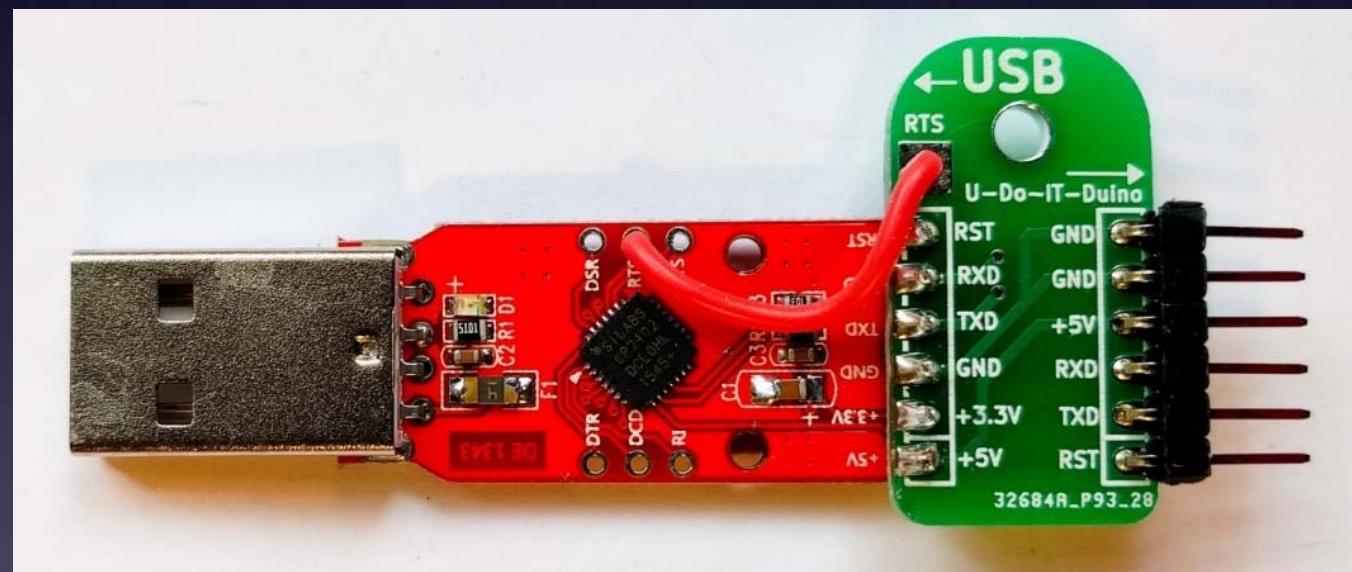
Old ones from Cornfield Electronics looked like this:



Connecting your ArduTouch to your computer

USB-Serial adapter cable

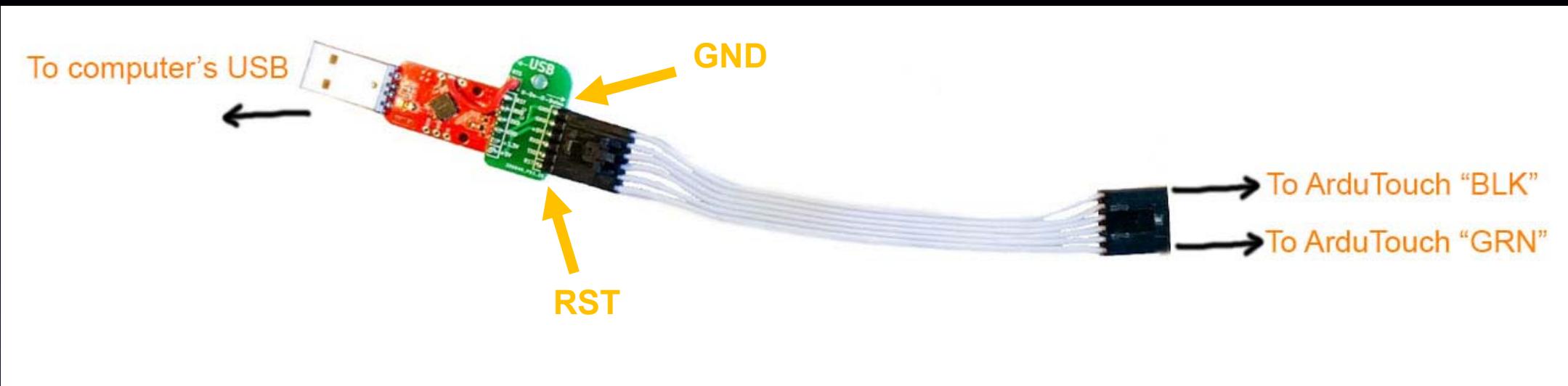
Old ones from Cornfield Electronics looked like this:



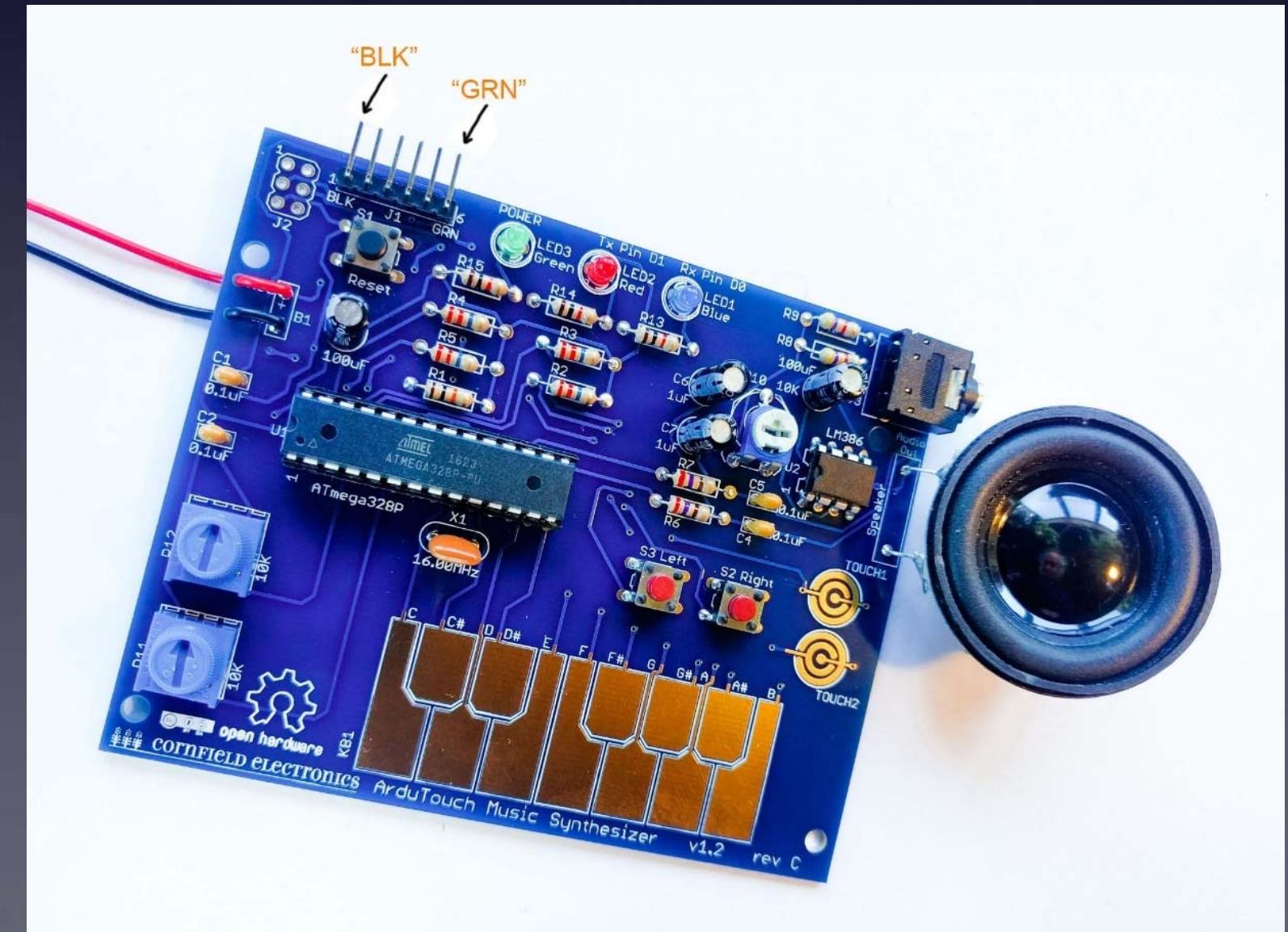
You may need to download and install a driver
for your Operating System (Windows, MacOS, or Linux):

[<https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers>](https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers)

Connecting your ArduTouch to your computer

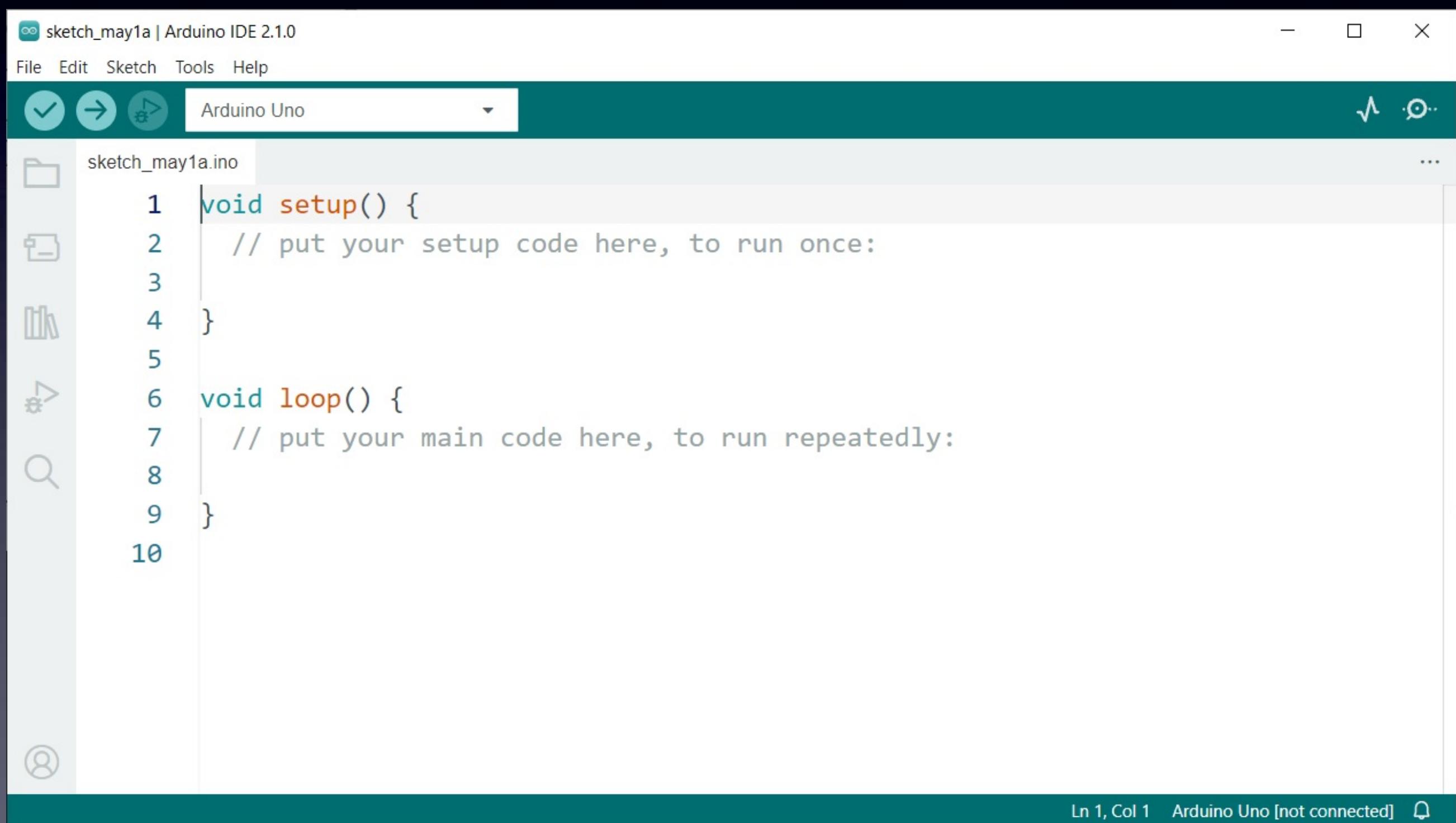


IMPORTANT:
Make sure the
battery pack on your
ArduTouch
is OFF



Arduino

After you download and install the Arduino software start it, and you will see a screen that looks like this:

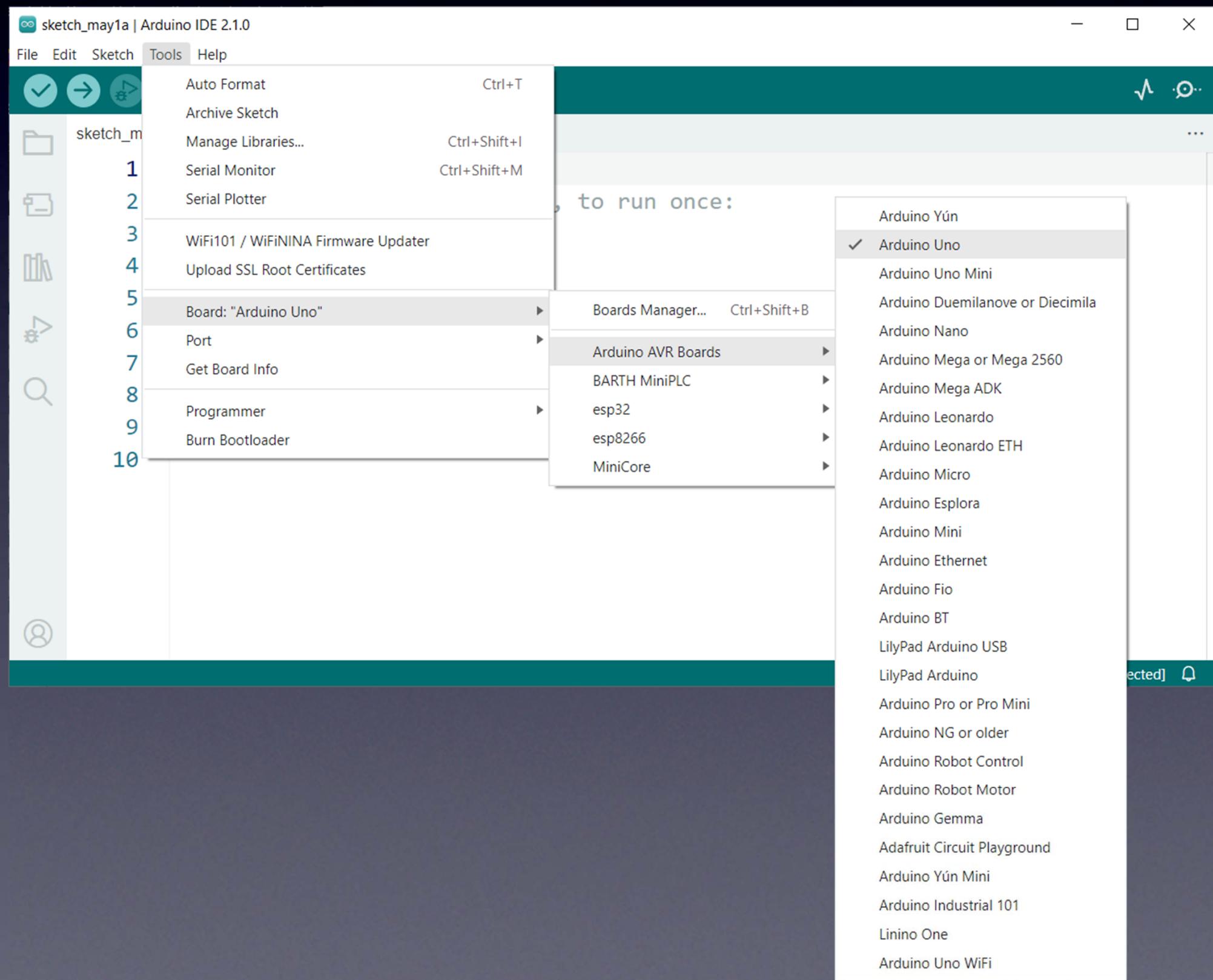


Arduino

The first time you start your Arduino software
you need to do three things to set things up

(1)
**Choose “Uno”
as the Board**

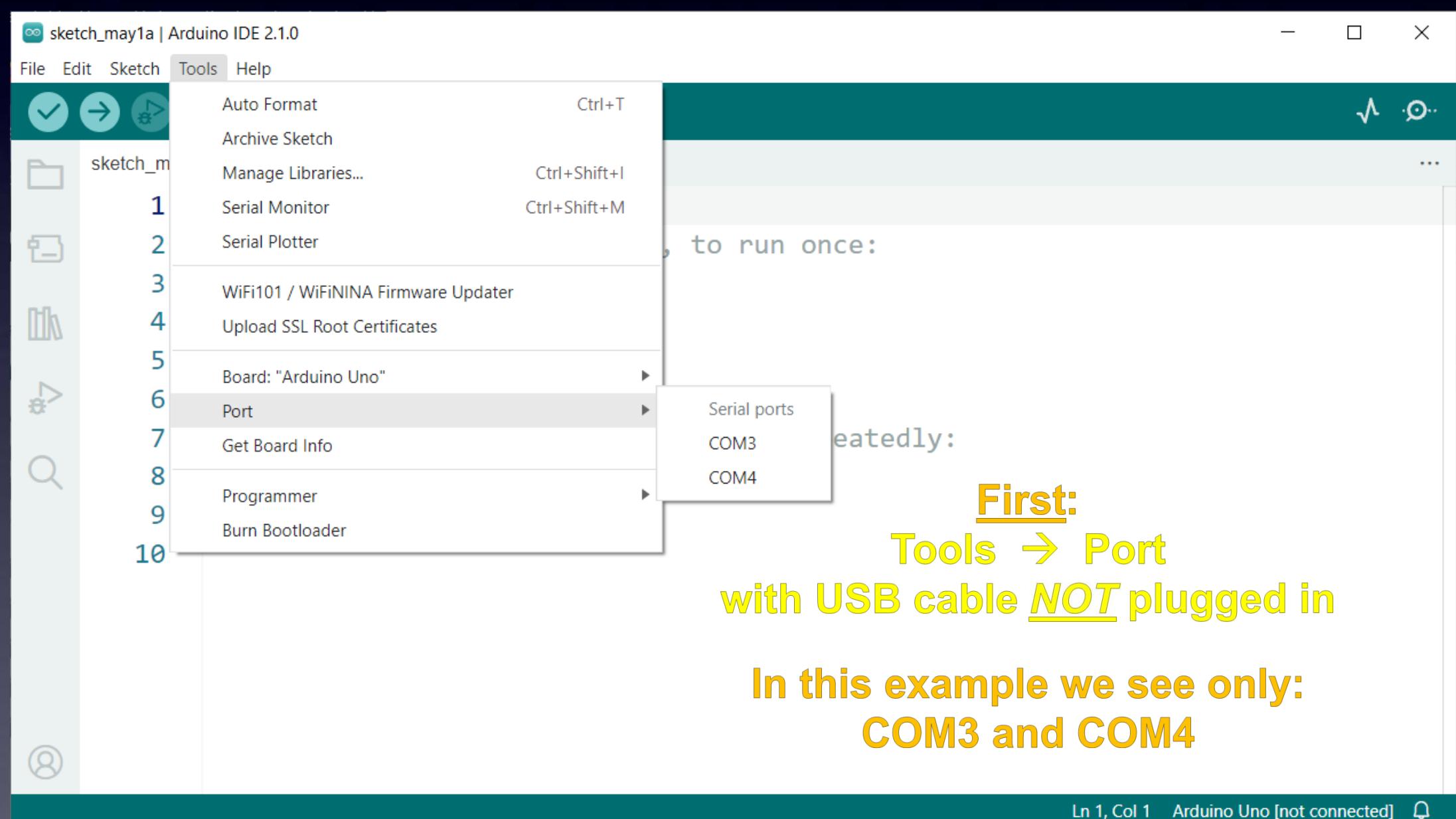
(Your
ArduTouch board
acts
just like
an
Arduino Uno board)



Arduino

The first time you start your Arduino software
you need to do three things to set things up

(2)
**Choose
the Port
(this will be
different
depending on
your Operating
System)**



First:
Tools → Port
with USB cable NOT plugged in

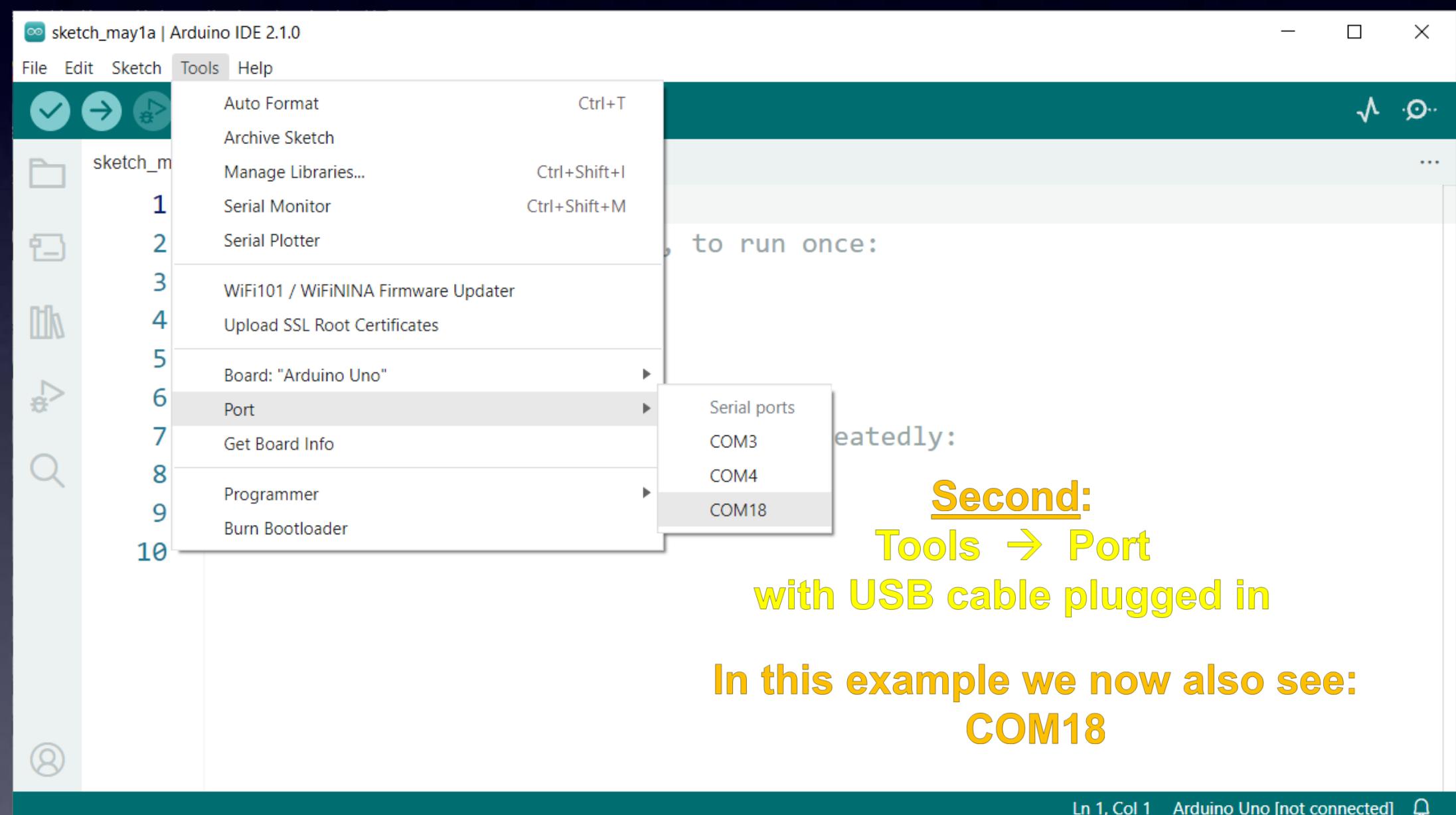
In this example we see only:
COM3 and COM4

Arduino

The first time you start your Arduino software
you need to do three things to set things up

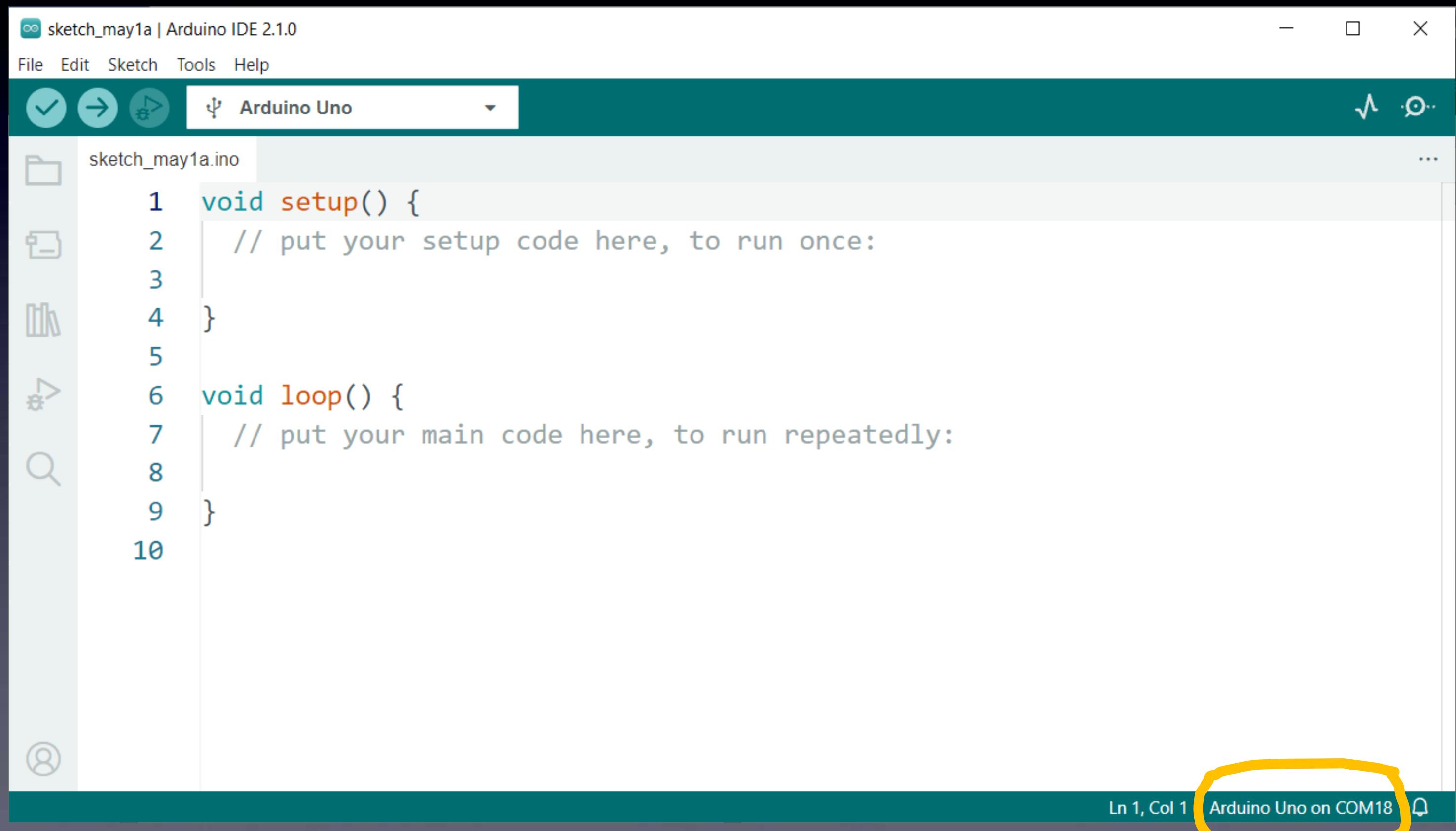
(2)
**Choose
the Port
(this will be
different
depending on
your Operating
System)**

(After installing
the driver for
your USB-Serial cable
and plugging it in
your operating system
will see a serial port
and it appears here.)



Arduino

Your Arduino software is almost ready



sketch_may1a | Arduino IDE 2.1.0

File Edit Sketch Tools Help

Arduino Uno

sketch_may1a.ino

```
1 void setup() {
2     // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7     // put your main code here, to run repeatedly:
8
9 }
10
```

Ln 1, Col 1 Arduino Uno on COM18

Arduino

The first time you start your Arduino software
you need to do three things to set things up

(3) Install the ArduTouch library



The screenshot shows a web browser window displaying the Cornfield Electronics website at <https://cornfieldelectronics.com/cfe/cfe.main.php>. The page features a yellow header with the text "cornFIELD electronics" and "useful electronics for a better world". Below the header, there are navigation links for "home", "buy", "about us", "press", "distributors", "projects", and "show cart". The main content area contains several images of electronic products: a blue Arduino Uno board, a white kit with components, a purple "Neuro Dreamer" sleep mask, and a black remote control. To the right, a large section titled "Take control" discusses the company's mission to empower users through their products like the TV-B-Gone and NeuroDreamer. Smaller text on the right side mentions the ArduTouch music synthesizer kit and TV-B-Gone kit.

Take control

At Cornfield Electronics we create devices that give people opportunities for effective choices in their lives. Each of us can decide whether to watch TV monitors, and when to watch. Each of us can decide when to get the rest we want, and how we dream. Everyone can learn to make cool things with our kits. Please explore our [products](#), make your own choices, and see how *your* life can be enhanced.

join our mailing list

Love it or hate it, TV screens are all around us. **TV-B-Gone®** universal remote control is the first fruit of our technical savvy, embodying our belief in empowerment, and sense of humor. This universal remote control fits in your pocket and allows you to discreetly turn TVs off wherever you go. TV-B-Gone fans around the world are using it for a variety of practical, philosophical, and humorous purposes. Imagine the possibilities...

Years in the making **NeuroDreamer** sleep mask is another of our personal empowerment inventions. We all need rest, but we don't always get it in our busy lives. NeuroDreamer sleep mask lets you use your own brainwaves to

bring you the rest you need. And with the **lucid dreaming model**, you can take control of your dreams.

Want to learn electronics? We make way cool, fun, intriguing, educational [kits](#) that **anyone can make!** Our most **POPULAR** kits are: [ArduTouch music synthesizer kit](#) and [TV-B-Gone kit!](#)

We make truly useful technological solutions that put you in charge.

Welcome to our better world!

NOTE: As of 14-Feb-2023 Cornfield Electronics is a sole proprietorship of Mitch Altman.

legal notices & privacy policy

CC BY-SA 2023 cornfield electronics

Arduino

The first time you start your Arduino software
you need to do three things to set things up

(3)
Install
the
ArduTouch
library



The screenshot shows the homepage of the Cornfield Electronics website. At the top, there's a navigation bar with links for File, Edit, View, History, Bookmarks, Tools, and Help. Below that is a browser header with tabs for 'Cornfield Electronics :: Home' and a search bar. The main content area features a yellow banner with the text 'cornFIELD electronics' and 'useful electronics for a better world'. Below the banner, there are several product images: a blue Arduino Uno board, a white kit with components, a purple 'Neuro Dreamer' sleep mask, and a black remote control. To the right, a large section is titled 'Take control' with a subtext about creating opportunities for effective choices. A green arrow points from the text 'you need to do three things to set things up' in the previous slide to the 'projects' link in the website's navigation bar.

At Cornfield Electronics we create devices that give people opportunities for effective choices in their lives. Each of us can decide whether to watch TV monitors, and when to watch. Each of us can decide when to get the rest we want, and how we dream. Everyone can learn to make cool things with our kits. Please explore our [products](#), make your own choices, and see how *your* life can be enhanced.

join our mailing list

Love it or hate it, TV screens are all around us. **TV-B-Gone®** universal remote control is the first fruit of our technical savvy, embodying our belief in empowerment, and sense of humor. This universal remote control fits in your pocket and allows you to discreetly turn TVs off wherever you go. TV-B-Gone fans around the world are using it for a variety of practical, philosophical, and humorous purposes. Imagine the possibilities...

Years in the making **NeuroDreamer** sleep mask is another of our personal empowerment inventions. We all need rest, but we don't always get it in our busy lives. NeuroDreamer sleep mask lets you use your own brainwaves to

bring you the rest you need. And with the **lucid dreaming model**, you can take control of your dreams.

Want to learn electronics? We make way cool, fun, intriguing, educational **kits** that **anyone can make!** Our most **POPULAR** kits are: [ArduTouch music synthesizer kit](#) and [TV-B-Gone kit!](#)

We make truly useful technological solutions that put you in charge.

Welcome to our better world!

NOTE: As of 14-Feb-2023 Cornfield Electronics is a sole proprietorship of Mitch Altman.

legal notices & privacy policy

CC BY-SA 2023 cornfield electronics

Arduino

The first time you start your Arduino software
you need to do three things to set things up

(3) Install the ArduTouch library



The screenshot shows a web browser window with the following details:

- Address Bar:** https://cornfieldelectronics.com/cfe/projects.php?PHPSESSID=d5d4714nuevrq25drkkoirr1m3
- Page Title:** cornFIELD electronics
- Page Description:** useful electronics for a better world
- Navigation:** home, buy, about us, press, distributors, projects, show cart
- Content Area:**
 - Section Header:** DO-IT-YOURSELF PROJECTS
 - Text:** by [Mitch Altman](#), and friends.
Last modified: 5-Oct-2022
 - Text:** You Can Make Cool Things With Electronics!
The projects on this page were all created for total beginners, with no experience,
so everyone can complete them successfully at my workshops, or at home, or anywhere!
 - Text:** All you need is:
a desire, a handful of parts, a soldering iron (with stand and sponge), a wire-cutter, a wire-stripper, solder, and an afternoon.
 - Image:** A photograph of basic electronic tools: a soldering iron, a roll of solder, a pair of pliers, and a pair of wire cutters.
 - Text:** [Here](#) is a really nice tutorial on how to solder -- for total beginners!
[Soldering Tutorial for total beginners](#)
 - Text:** Open Hardware!
Everything on this page (and everything I do) is free and open source!
(That's *free* as in *freedom*.)
(But everything here is free to download -- and that is *free* as in *beer*.)
 - Text:** If you have any questions on anything, please feel free to email me:
Transferring data from cornfieldelectronics.com... [ch AT CornfieldElectronics DOT com](mailto:)

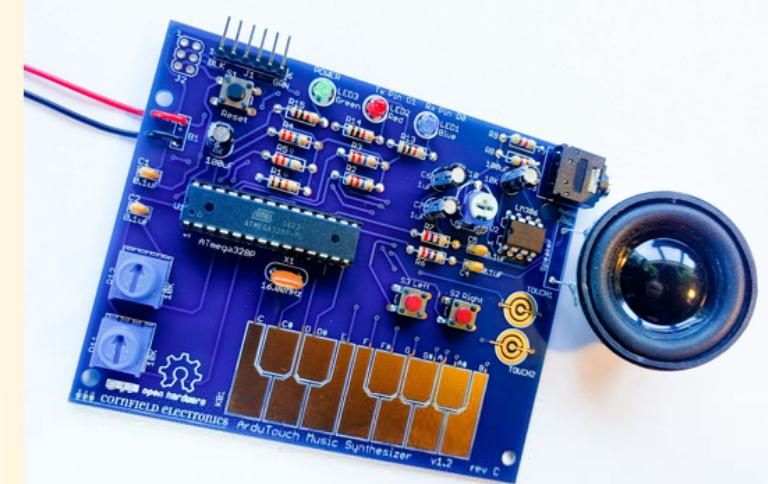
Arduino

The first time you start your Arduino software
you need to do three things to set things up

(3) Install the ArduTouch library



Project: ArduTouch Arduino-compatible Music Synthesizer kit
-- make way cool sounds and music!



==> [BUY an ArduTouch music synthesizer kit! <==](#)

Solder your ArduTouch kit together, and it works! You can make way wonderful music, sound, and noise. Use the ArduTouch Library or hack the existing sketches to create your own cool synthesizers. *The documentation is getting good enough to learn how to use Digital Signal Processing (DSP) to make your own sounds for your own projects. (More documentation coming.)*

This kit takes about 120 minutes to complete.

For assembly instructions, please see:
[ArduTouch assembly instructions for Rev C board](#)

older versions (before 2017):
[\(assembly instructions for Rev B, Rev A, and mono\)](#)

To program your ArduTouch music synthesizer kit, you'll need a USB-Serial TTL cable, such as an FTDI Friend or FTDI Cable, available all over the place. You can [purchase a nice one](#) from Cornfield Electronics. These USB-Serial TTL cables (made by Samurai Circuits), require a driver (from Silicon Labs):
[Samurai Circuits board \(SiLabs CP210x USB-to-Serial TTL\) drivers:](#)
[The latest drivers from Silabs' website \(Windows, MacOS, Linux\)](#)

You will also need to download the free, open-source, Arduino software (for Windows, MacOS, or Linux).
[Arduino software](#) (the latest version is fine to use).

Scroll down

Arduino

The first time you start your Arduino software
you need to do three things to set things up

(3) Install the ArduTouch library

Here are the slides I use when I give my ArduTouch workshops
(including assembly instructions and reprogramming instructions):
[Slides I use for ArduTouch workshops.](#)

One ArduTouch synth is really nice -- and 7 ArduTouch synths are even nicer!
Here's a demo Video:

The ArduTouch library and example sketches will work on any Arduino board!
The ArduTouch board behaves like an Arduino Uno.

The ArduTouch Library contains everything you need to start creating your own synthesizers! It was mostly written by my friend Bill Alessi. The ArduTouch Library comes with a sequence of **example sketches** -- read through these and try them! As well as being way cool synthesizers, they also serve as **really good tutorials** on how to create your own synthesizer sketches for ArduTouch. You can download the ArduTouch library, and then import it using the Arduino software:
[ArduTouch Library v1.16](#) for ArduTouch Music Synthesizer kit.

Thick is an example of a **way cool, easy-to-play synthesizer sketch** for ArduTouch! Check it out -- it sounds like it comes from a vintage analog Mini Moog. (Your ArduTouch synthesizer kit comes pre-programmed with this synthesizer.)
[Instructions for how to use the Thick synthesizer.](#)
Thick was written by my friend Bill Alessi. The sketch will work on any Arduino (the ArduTouch has its own Arduino Uno clone built in). Thick uses the ArduTouch library (so be sure to download it, too -- see above).
[Thick v0.72 synthesizer sketch](#)

Thick Demo Video:

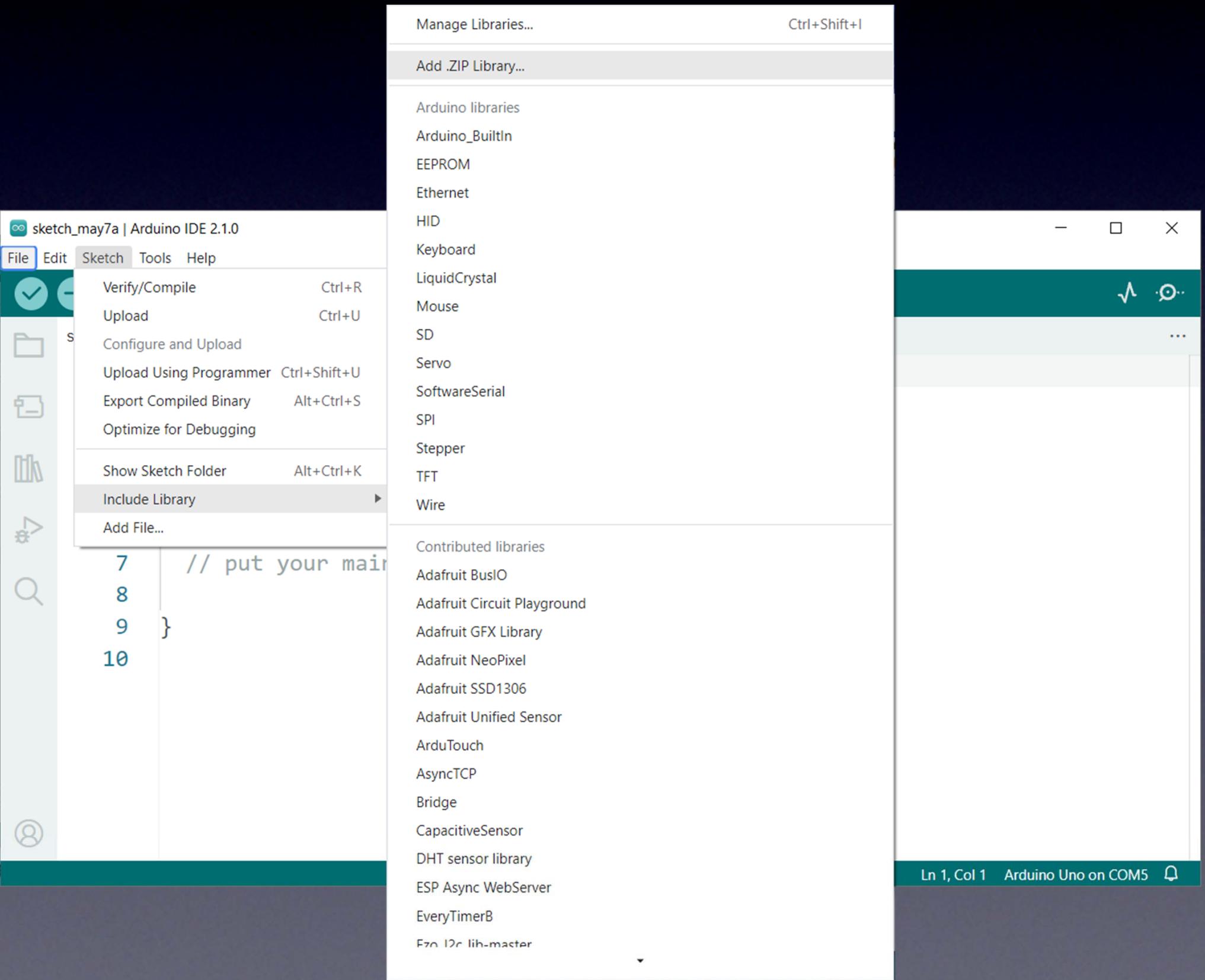
Scroll down

click this link to download the ArduTouch library

Arduino

The first time you start your Arduino software
you need to do three things to set things up

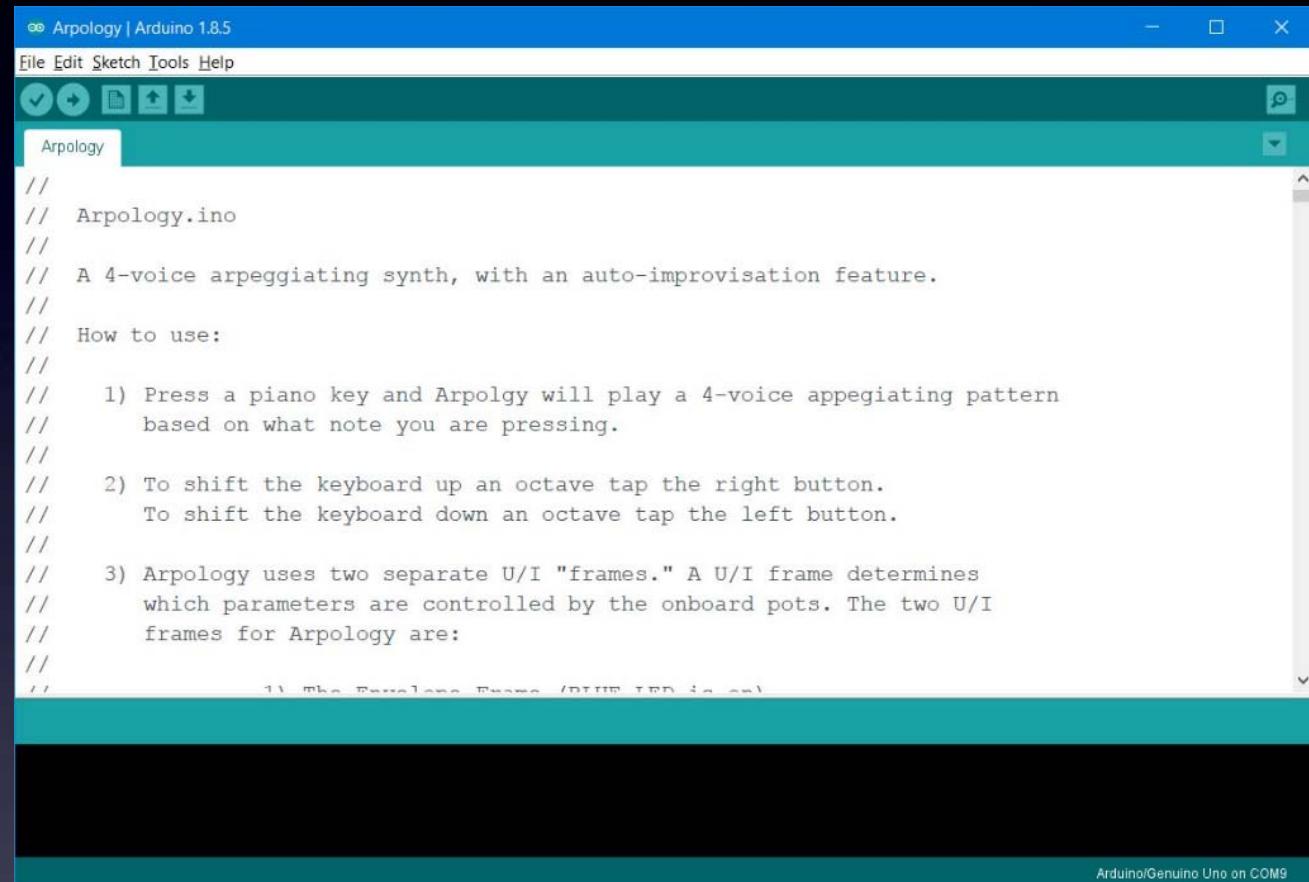
(3)
**Install
the
ArduTouch
library**



Arduino

Designed for non-geeky artists

Download
a new
ArduTouch
synth “sketch”



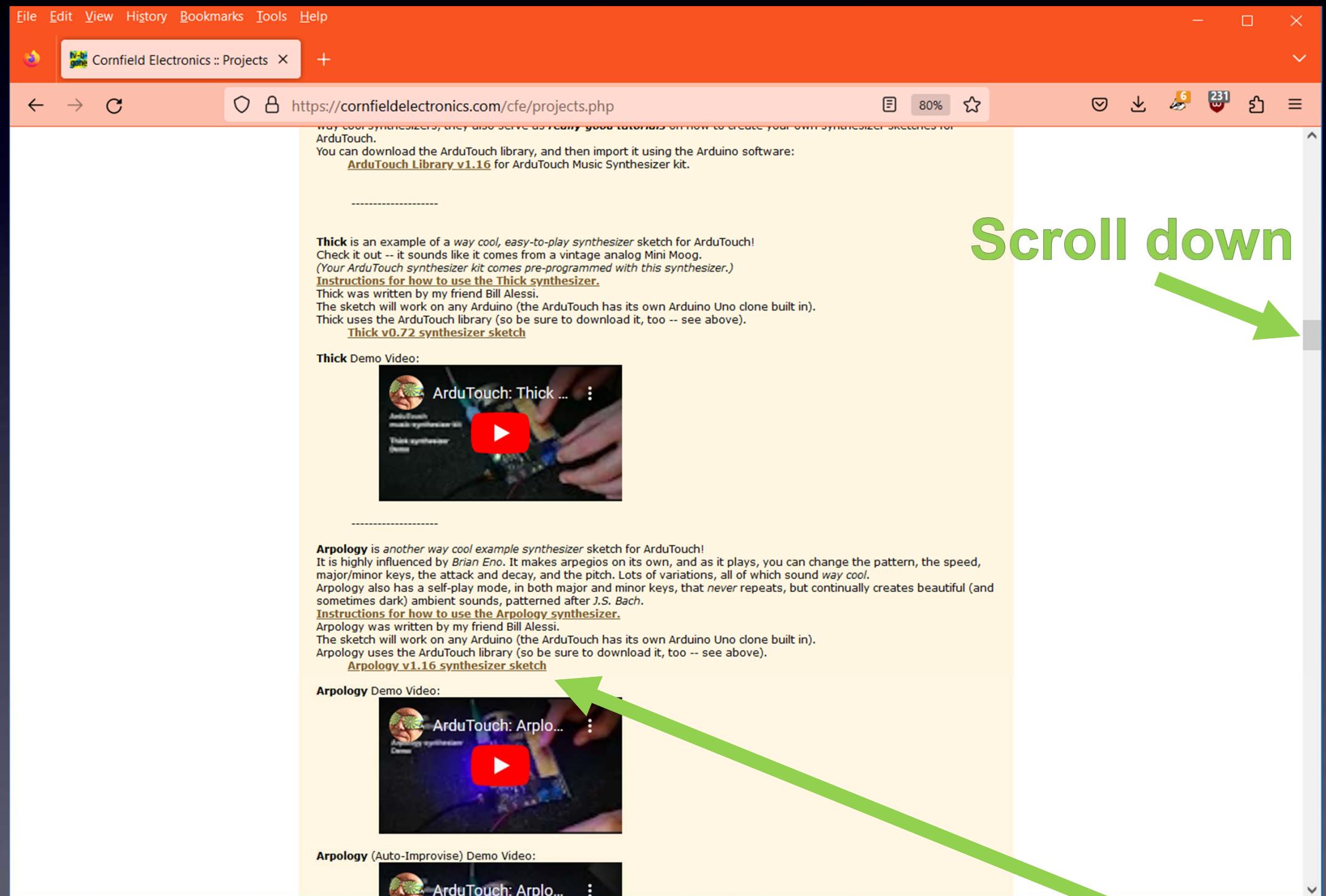
The screenshot shows the Arduino IDE interface with a sketch titled "Arpology.ino". The code is a C++ program for an Arduino Uno. It includes comments explaining the purpose of the sketch (a 4-voice arpeggiating synth), how to use it, and instructions for interacting with a keyboard. The code also mentions two separate UI frames controlled by onboard pots. At the bottom of the code editor, there is a status bar indicating "Arduino/Genuino Uno on COM9".

```
//  
// Arpology.ino  
//  
// A 4-voice arpeggiating synth, with an auto-improvisation feature.  
//  
// How to use:  
//  
// 1) Press a piano key and Arpolgy will play a 4-voice appiegiating pattern  
// based on what note you are pressing.  
//  
// 2) To shift the keyboard up an octave tap the right button.  
// To shift the keyboard down an octave tap the left button.  
//  
// 3) Arpology uses two separate U/I "frames." A U/I frame determines  
// which parameters are controlled by the onboard pots. The two U/I  
// frames for Arpology are:  
//  
// 11. The Parallel Frame (DUE, TEE, 4c, etc)
```

“Sketch” :
an Arduino program

Arduino

Download a new ArduTouch synth “sketch”



click link to download a synth “sketch”

Arduino

Download a new ArduTouch synth “sketch”

Check it out!

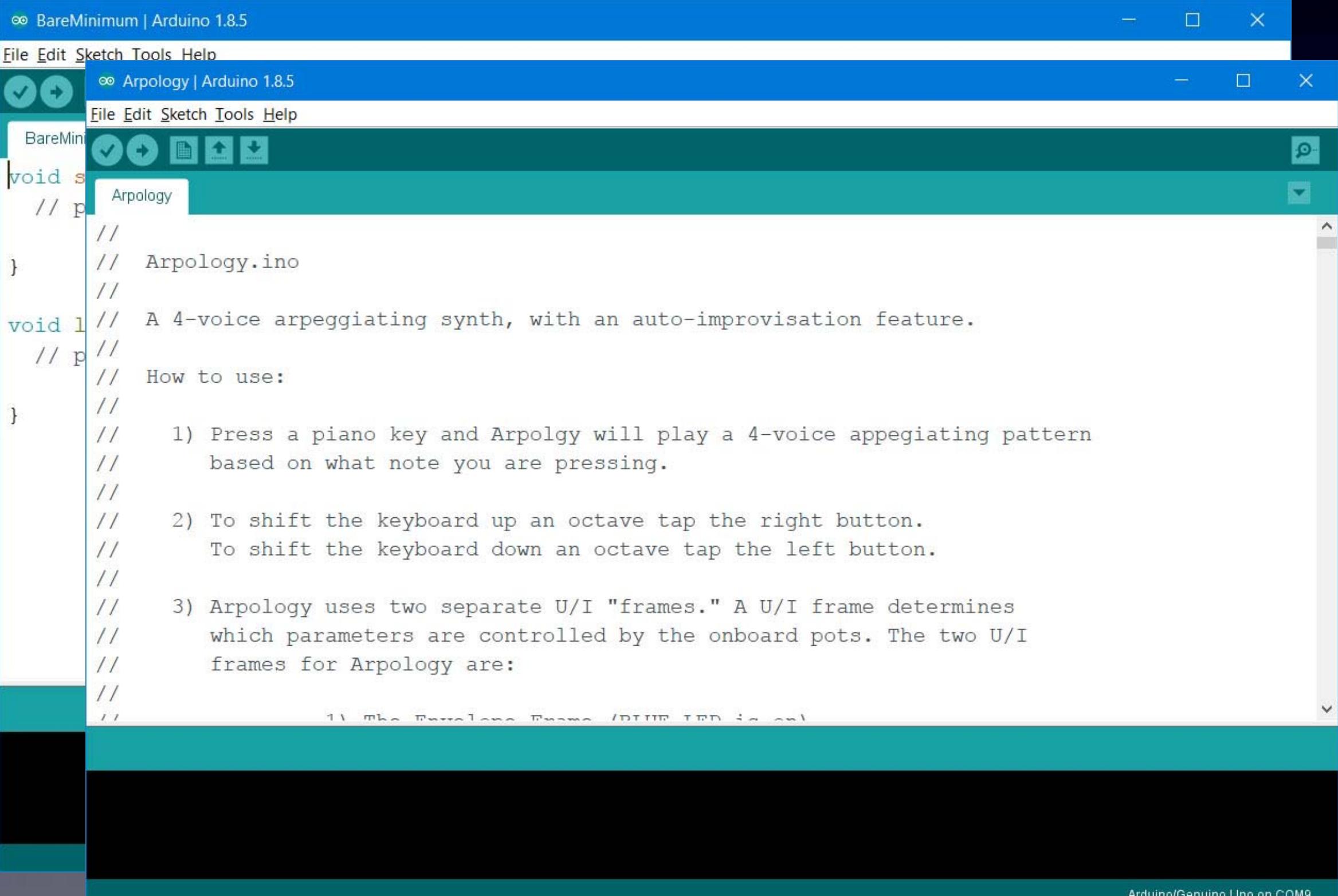
**Also
available
for each synth:**

- Demo Videos
- Instructions

Arduino

You can open the ArduTouch synth sketch:
File → Open...

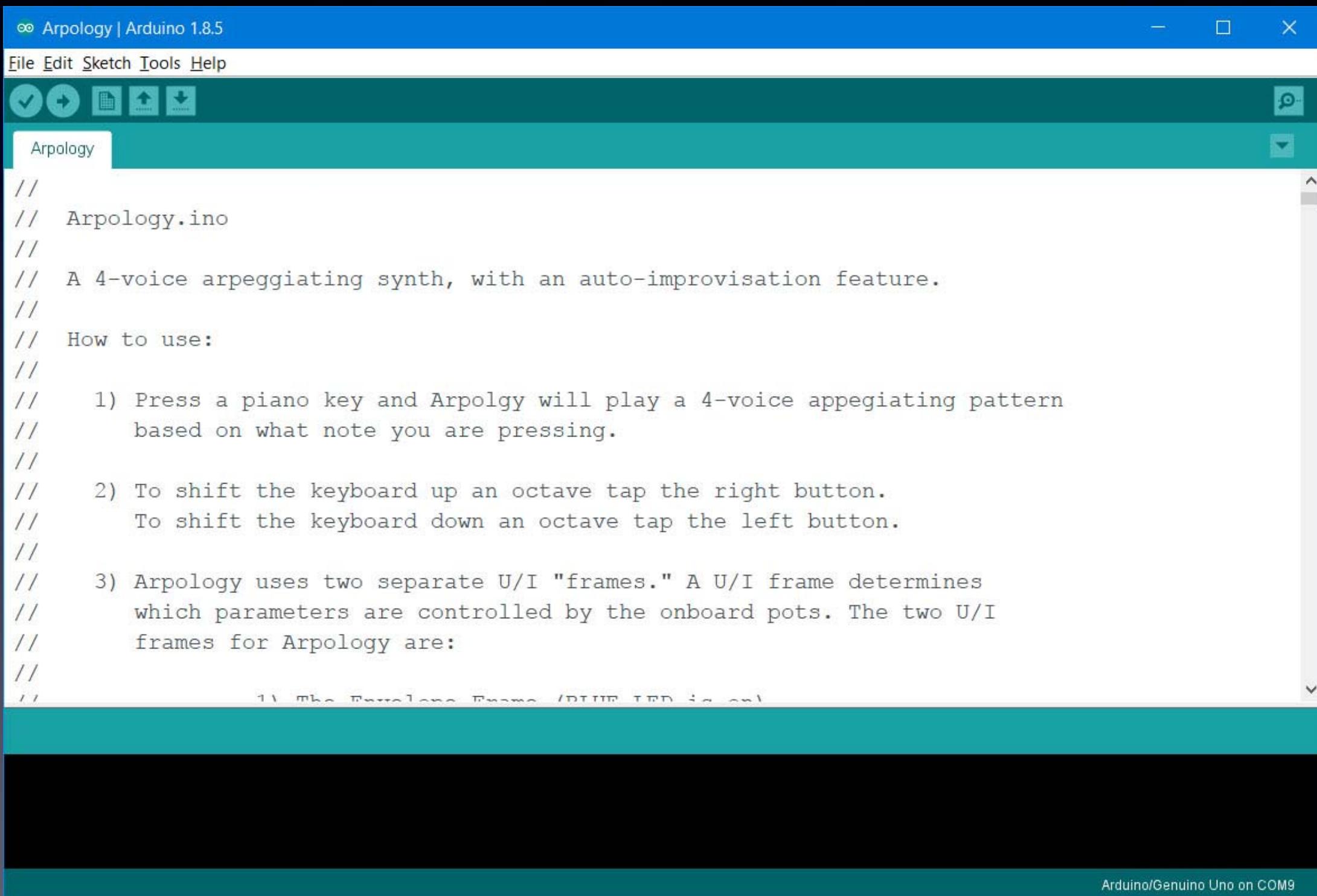
(I opened “Apology here)



```
BareMinimum | Arduino 1.8.5
File Edit Sketch Tools Help
Arpology | Arduino 1.8.5
File Edit Sketch Tools Help
BareMinimum.ino
void setup() {
    // put your setup code here, to run once:
    // Arpology
}
// Arpology.ino
//
void loop() {
    // A 4-voice arpeggiating synth, with an auto-improvisation feature.
    // How to use:
    // 1) Press a piano key and Arpolgy will play a 4-voice appegiating pattern
    //     based on what note you are pressing.
    //
    // 2) To shift the keyboard up an octave tap the right button.
    //     To shift the keyboard down an octave tap the left button.
    //
    // 3) Arpology uses two separate U/I "frames." A U/I frame determines
    //     which parameters are controlled by the onboard pots. The two U/I
    //     frames for Arpology are:
    //
    //     1) mba_PianoKeys_Piano / DTUNE TPD is on!
```

Arduino

You can now program your ArduTouch with a new synth sketch !



The screenshot shows the Arduino IDE interface with the title bar "Arpology | Arduino 1.8.5". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for back, forward, file operations, and a search function. The main window displays the code for the "Arpology" sketch. The code is a C++ program with comments explaining its functionality. It describes a 4-voice arpeggiating synth with an auto-improvisation feature. It details how to use the device by pressing piano keys, shifting octaves with right and left buttons, and using two UI frames controlled by onboard pots. The code ends with a note about the "Envelopes Frame" and the state of the blue LED.

```
//  
// Arpology.ino  
//  
// A 4-voice arpeggiating synth, with an auto-improvisation feature.  
//  
// How to use:  
//  
// 1) Press a piano key and Arpolgy will play a 4-voice appegiating pattern  
//     based on what note you are pressing.  
//  
// 2) To shift the keyboard up an octave tap the right button.  
//     To shift the keyboard down an octave tap the left button.  
//  
// 3) Arpology uses two separate U/I "frames." A U/I frame determines  
//     which parameters are controlled by the onboard pots. The two U/I  
//     frames for Arpology are:  
//  
//     1) The Envelope Frame (BLUE LED is on)  
//  
//     2) The Modulation Frame (BLUE LED is off)
```

Arduino/Genuino Uno on COM9

Arduino

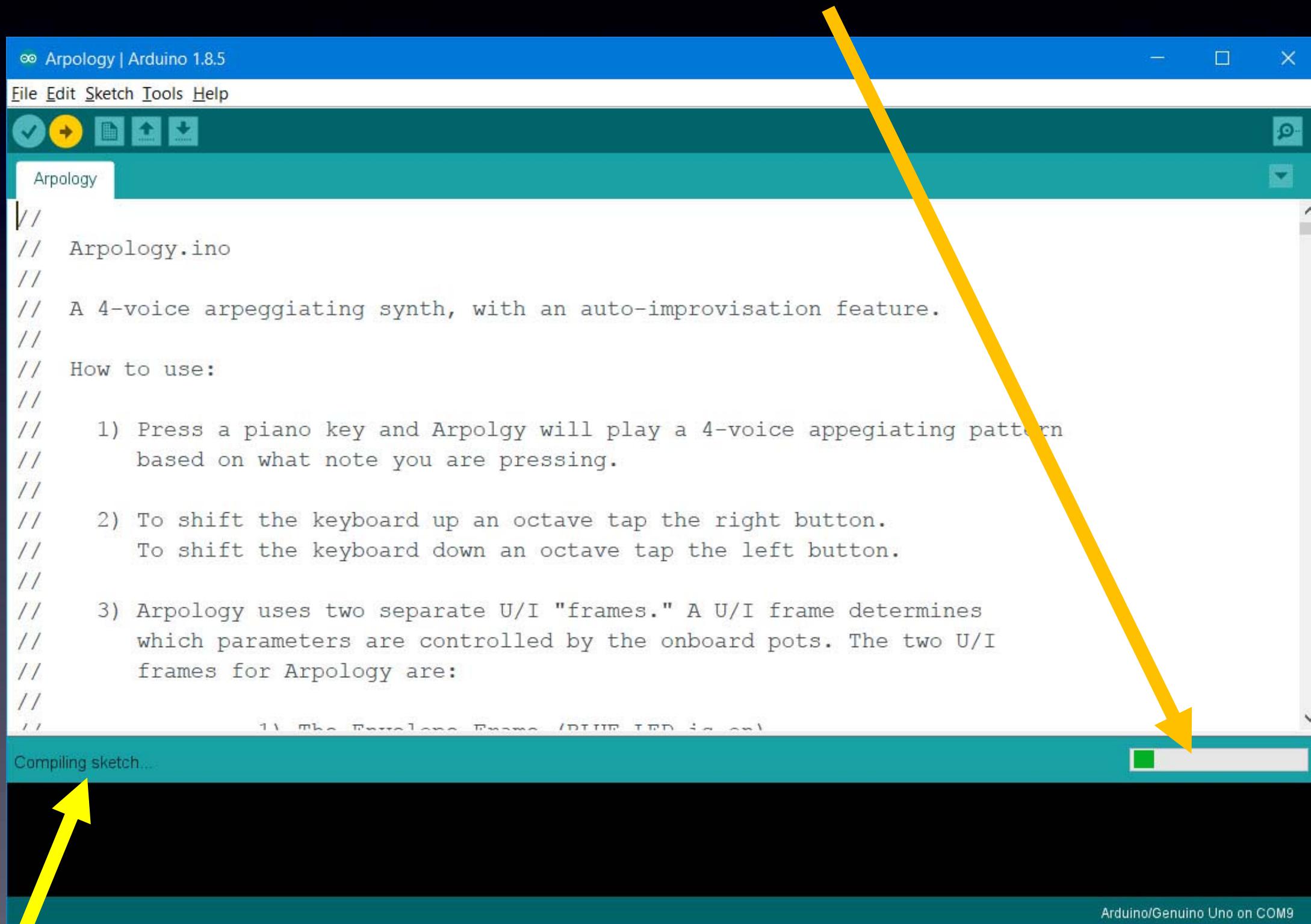
**With the USB-Serial cable connected to your ArduTouch board
press the Upload button**



```
Arpology | Arduino 1.8.5
File Edit Sketch Tools Help
Arpology
// 
// Arpology.ino
//
// A 4-voice arpeggiating synth, with an auto-improvisation feature.
//
// How to use:
//
// 1) Press a piano key and Arpolgy will play a 4-voice appegiating pattern
//    based on what note you are pressing.
//
// 2) To shift the keyboard up an octave tap the right button.
//    To shift the keyboard down an octave tap the left button.
//
// 3) Arpology uses two separate U/I "frames." A U/I frame determines
//    which parameters are controlled by the onboard pots. The two U/I
//    frames for Arpology are:
//
//    1) The Envelope Frame (BLUE LED is on)
Compiling sketch...
Arduino/Genuino Uno on COM9
```

Arduino

While uploading, you will see a progress bar...



...and when it's completed successfully, it says: “Upload done”

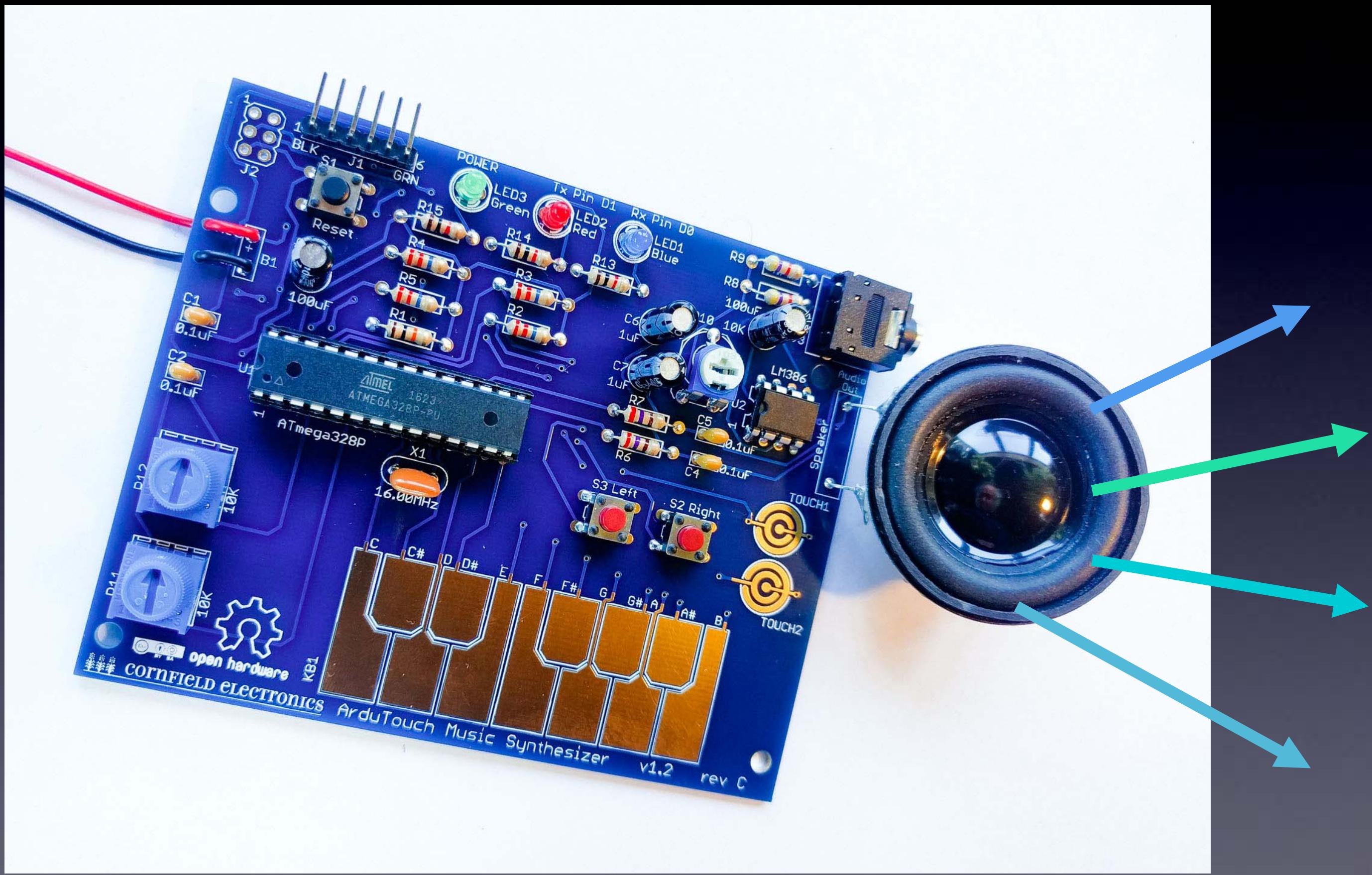
ArduTouch

**Disconnect your ArduTouch board
from the USB-Serial cable,**

turn on your battery pack,

And...

Let's make new noise!



Please Remember:

to

Wash your hands
after soldering

Learn to Solder with ArduTouch Music Synthesizer kit and make music, sound, and noise!

Mitch Altman

Chief Scientist, **Cornfield Electronics**, San Francisco, CA

Inventor of **TV-B-Gone** universal remote controls

Co-founder of **3Ware** (successful Silicon Valley startup)

Pioneer of **VR** (in the mid-1980s)

Founding mentor at **HAX** (1st and biggest hardware accelerator)

Co-founder of **Noisebridge** (San Francisco hackerspace)

email: mitch@CornfieldElectronics.com

site: www.CornfieldElectronics.com

facebook: [maltman23](https://www.facebook.com/maltman23)

flickr: [maltman23](https://flickr.com/photos/maltman23)

WeChat: [mitchaltman](https://wechat.com/mitchaltman)

Fediverse: @maltman23@mastodon.social

Patreon: [mitchaltman](https://www.patreon.com/mitchaltman)



cornFIELD electronics