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)

e-mail: mitch@CornfieldElectronics.com
site: www.CornfieldElectronics.com
twitter: @maltman23
flickr: maltman23
WeChat: mitchaltman
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

---

7-Oct-2020
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
Digital Signal Processing

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

This is called: “Sampling”
Interactive Digital Signal Processing:

To record it digitally:

- First slice it
- Then get the values
- Then store the values

Waveform is: "Digitized"

Samples stored in Memory:

<table>
<thead>
<tr>
<th>Memory loc</th>
<th>Memory contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.0</td>
</tr>
<tr>
<td>1</td>
<td>7.8</td>
</tr>
<tr>
<td>2</td>
<td>11.9</td>
</tr>
<tr>
<td>3</td>
<td>10.5</td>
</tr>
<tr>
<td>4</td>
<td>4.0</td>
</tr>
<tr>
<td>5</td>
<td>-4.0</td>
</tr>
<tr>
<td>6</td>
<td>-10.3</td>
</tr>
<tr>
<td>7</td>
<td>-11.9</td>
</tr>
<tr>
<td>8</td>
<td>-7.8</td>
</tr>
<tr>
<td>9</td>
<td>0.0</td>
</tr>
</tbody>
</table>
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
Digital Signal Processing

Square Wave:
ON half the time / OFF half of the time
Digital Signal Processing

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

Pulse Wave:
ON and OFF at any ratio you like
Digital Signal Processing

Pulse Width Modulation (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:

Sound Waveform “Envelope”
Digital Signal Processing

Some “Dynamics”:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ADSR</td>
<td></td>
</tr>
<tr>
<td>Tremolo – <em>constant changing volume</em></td>
<td></td>
</tr>
<tr>
<td>Portamento</td>
<td></td>
</tr>
<tr>
<td>Envelopes</td>
<td></td>
</tr>
<tr>
<td>Filters</td>
<td></td>
</tr>
<tr>
<td>Effects</td>
<td></td>
</tr>
</tbody>
</table>
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

With Tutorial examples

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

// the following line is required for every ArduTouch sketch

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

void loop()
{
    arduTouch_loop();  // perform ongoing ArduTouch tasks
ArduTouch
Arduino-Compatible

With Tutorial examples
Follow examples 01 through 09 to easily learn to code your own synthesizers
ArduTouch
Arduino-Compatible

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

Complete code for:
- sawtooth waves
- play with keyboard
- change octaves
- volume control
ArduTouch

Easy to add:
- Tremolo
- Portamento
- Envelopes
- Filters
- Effects
- Other waveforms
ArduTouch

Open Hardware – everything is on Github

maltman23
Tools

(Don’t bring these home)
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.)
SOLDERING IS EASY
HERE’S HOW TO DO IT

BY: MITCH ALTMAN
(SOLDERING WISDOM)
ANDIE NORDGREN
(COMICS ADAPTATION)
JEFF KEYZER
(LAYOUT AND EDITING)

DOWNLOAD THIS COMIC BOOK AND SHARE IT WITH YOUR FRIENDS!
HTTP://MIGHTYOHM.COM/SOLDERCOMIC

DISTRIBUTE WIDELY!

Download for free at:
http://mightyohm.com/soldercomic

Learn To Solder
Learn To Solder

Download in the language of your choice for free at: http://mightyohm.com/soldercomic
SOLDER C’EST FACILE
VOICI COMMENT FAIRE

DE: MITCH ALTMAN
(MAÎTRE SOUDEUR)

ANDIE NORDGREN
(ADAPTATION BD)

JEFF KEYZER
(ÉDITION, MISE EN PAGE)

SNOOTLAB
(TRADUCTION FR.)

TÉLÉCHARGEZ CETTE BD
ET PARTAGEZ LA AVEC VOS AMIS !
HTTP://MIGHTYOHM.COM/SOLDERCOMIC

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

Some parts are inside of this battery pack
The board we’ll solder the parts to
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

**Important:** Use solder WITH lead (Pb)!! Unleaded solder has very poisonous fumes!
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 – 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)
The perfect kind of solder for electronics:

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

(63/37 is also good)

Important:
Use solder WITH lead (Pb) !!
lead-free solder has very poisonous fumes!
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
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!*
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

Push solder in, over the pad, and under the tip

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

Push in about 1 mm of solder

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
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
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 proper orientation
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: 1μF
C3, C8: 100uF
C3, C8: Long Lead “+”

Use 100uF !!
C3, C8: 100uF – soldered to board
C6, C7: 1μF
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
J1

long leads

short leads
Short leads into board

short leads go into the board

→ long leads sticking out from board
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.
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.
When chips are new, their pins are bent out.
We need the pins bent straight and parallel. Use your work table to (gently) bend the leads.
Gently bend leads so they're straight and parallel

U2
U2: audio amp chip

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

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.
U1: microcontroller
These pins must be straight and parallel
U1: microcontroller
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
Tin one side of each lead
(i.e., cover with thin film of melted solder)
Solder one lead to speaker

Notice the correct place to solder the wire
Solder next lead to speaker

Notice the correct place to solder the wire

Speaker
Some kits have a speaker that looks like this.

Notice the correct place to solder the wires.
Insert speaker into board and solder both leads to board.
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!
Loop one lead into its pad, and solder.
Then loop the other lead into its pad, and solder.
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>
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:
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

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>

(details on this soon)
Re-programming the ArduTouch

Third:
Download ArduTouch synth sketches
<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:
Connecting your ArduTouch to your computer

USB-Serial adapter cable

Ones available from Cornfield Electronics look like this:

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):

Connecting your ArduTouch to your computer

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

(Old ones)
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.

1. Choose the Port (this will be different depending on your Operating System).

2. 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:

1. Choose the Port (this will be different depending on your Operating System).

2. 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.

Second: Tools → Port with USB cable plugged in.

In this example we now also see: COM18.
Arduino

Your Arduino software is almost ready

```cpp
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```
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

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

(3) Install 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

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

(3) Install the ArduTouch library
The first time you start your Arduino software you need to do three things to set things up:

1. Install the Arduino IDE
2. Connect your Arduino board
3. Install the ArduTouch library

Click this link to download the ArduTouch library:
https://comfieldelectronics.com/life/projects.php?PHPSESSID=d1c564714a4e5359957d1b1f6a2d

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 Axon.
The ArduTouch Library comes with a sequence of example sketches, 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:

Click this link to download the ArduTouch library:
https://comfieldelectronics.com/life/projects.php?PHPSESSID=d1c564714a4e5359957d1b1f6a2d

Click this link to download the ArduTouch library:
The first time you start your Arduino software you need to do three things to set things up:

1. Install the Arduino IDE
2. Connect your Arduino board to your computer
3. Install the ArduTouch library
Arduino

Designed for non-geeky artists

Download a new ArduTouch synth “sketch”

“Sketch”:

an Arduino program
Arduino

Download a new ArduTouch synth “sketch”
Arduino

Download a new ArduTouch synth “sketch”

Check it out!

Also available for each synth:
- Demo Videos
- Instructions
You can open the ArduTouch synth sketch:
File → Open...

(I opened “Arpology here)
You can now program your ArduTouch with a new synth sketch!
Arduino

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

```c
// Arpology
// A 4-voice arpeggiating synth, with an auto-improvisation feature.
// How to use:
// 1) Press a piano key and Arplogy 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) Arplogy 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 Arplogy are:

   1) The Envelope Frames (RUP, LUP to UE)
```

Compiling sketch...
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
twitter: @maltman23
flickr: maltman23
WeChat: mitchaltman