TeknoDida 2015
17.–19.4.2015

TeknoDida 2015 | Perjantai | Lauantai | Sunnuntai

Frank Clark
Chair and Professor
School of Music
Georgia Institute of Technology
fclark@music.gatech.edu

MUSIC TECHNOLOGY OUTREACH

Georgia Tech Center for Music Technology

Music Technology Outreach

- EarSketch
- Moog Werkstatt

EarSketch

Jason Freeman
Associate Professor of Music

earsketch.gatech.edu

Please Play:

Using HipHop to Teach Computer Science Video

EarSketch

computational music remixing and sharing as a tool to drive engagement and interest in computing

Contents

- Welcome
- Unit 1: Getting Started
- Unit 2: Effects and Beats
- Unit 3: For Loops, User-Defined Functions, Debugging
- Unit 4: Lists, Effects in More Detail, Musical Form
- Unit 5: Randomness
- Unit 6: Sonification
- Unit 7: Teaching Computers
- Unit 8: Creating effects
- Unit 9: Recursion
- Reference
  - Recording
  - Creating Beats with makeBeat()
  - EarSketch API
  - EarSketch Sound Library
  - Every Effect Explained in Detail
  - Every Analysis Feature Explained in Detail
  - Python Reference
  - Copyright
- Additional Examples
  - Creating a "Fractalized Arrangement"
  - Effect Case Studies
- Teaching Resources
  - Teacher Materials
  - Social Media Teaching Resources
- Teacher Curriculum
  - Teacher Module 1: An Overview of Musical Concepts
  - Teacher Module 2: Rhythms
  - Teacher Module 3: Pattern and Variety
  - Teacher Module 4: Effects
  - Teacher Module 5: FAQ
- Old Curriculum

Welcome

With EarSketch, you can learn computer science and music technology side by side. You will use Python, one of the most popular computer programming languages in the world, to create and remix music within the same kind of digital audio workstation (DAW) software used throughout the music industry. Along the way, we'll cover important computer science concepts like variables, functions, lists, loops, and conditionals, and we'll connect them to important music and music technology concepts like multi-track editing, effects, rhythm, and musical form.

Writing your own computer programs to create music might seem strange at first, but it's actually been an important part of the music industry since the earliest days of computers over 50 years ago. Musicians and computer scientists have always been writing computer code to try to create the next cool sound or effect or musical structure (from sci-fi soundtracks to dubstep grooves) and to design new ways to create and perform music (from intelligent computer musicians to unusual new musical instruments to iPhone apps). Throughout this curriculum, we'll show you ways in which writing code can help you more easily make music and make music that's more unique to you. And once you learn to write computer code, you can take those skills with you to any career you can imagine, whether in the music industry or elsewhere.

One of our EarSketch teachers, Darryl McCune, wanted to share his thoughts about EarSketch and computing with you. Watch his video below.

Download the video here if you aren't able to access it via Youtube

As you work through the EarSketch curriculum, you will need to use the EarSketch code editor and digital audio workstation environment to write code and make music. It is available **here** and runs inside of your web browser with the latest versions of Chrome, FireFox, or Safari. (Internet Explorer is not supported.)

Happy programming / writing music!

| Category: Uncategorized, Welcome | **Comments are closed**

1.2. EarSketch Social media site

(Return to Top)

The EarSketch social media site is a place for you to share your EarSketch projects with other students, friends and family. We encourage you to browse through existing projects, listen to them, and like, tag and comment on them. You may even find that these projects serve as inspiration for your own work. The rest of this section will walk you through creating an account on the website and take you on a tour of a few areas.

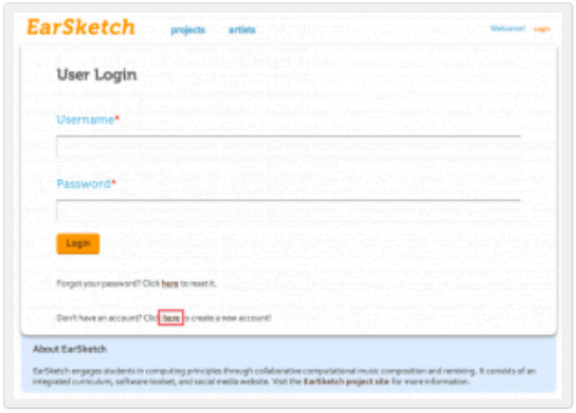
Creating an account

(Return to Top)

The first thing you will do is create your own account on the website by clicking this link: **Earsketch Social Media Website**

Social Media Website Login

To get started, you need to create an account on the site. Click the login link.



Social Media Website Account Creation Link

Exercise

First, let's add a very common effect to change the volume of a track. Paste the following code into our completed script from **Unit 1** at line 37:

```
37 #effects
38 setEffect(3, VOLUME, GAIN, 12.0)
```

Run the script. Do you notice what changed?

Track Number

The first parameter of the `setEffect()` function is the track number for which the effect will be applied. If we change the first parameter in this statement to a `4` instead of a `3` (so that we have `setEffect(4, VOLUME, GAIN, 12.0)`), the effect will go on track 4 when we load the script. Try this and listen but don't forget to change the value back to `3` afterward.

Effect Type

The second and third parameters of the `setEffect()` function specify the name of the effect and the parameter of the effect that is being applied to the track. The **Effects Reference** lists each and every effect and its parameters. Let's look at line 38 again:

```
38 setEffect(3, VOLUME, GAIN, 12.0)
```

In this line, the second and third parameters of `setEffect()` are `VOLUME` and `GAIN`. The first, `VOLUME`, is the name of the effect to be applied and the second, `GAIN`, is the name of the effect parameter. Every effect has a parameter that determines exactly what about the effect is being set. For example, `VOLUME` has two possible parameters, `GAIN` and `BYPASS` (see the effects reference for details). Another effect, PAN, has an effect parameter of LEFT_RIGHT, which indicates that the sound should pan from the left speaker to the right. Try applying a PAN effect on line 38 instead of the `VOLUME` effect:

```
38 setEffect(1, PAN, LEFT_RIGHT, -100)
```

If you're wearing headphones or have a stereo speaker setup, you will hear the difference in the track 1 melody, which is now panned all the way to the left of the stereo field (due to -100 effect value). The range of the value is between -100 and 100. Try changing it to different values to see how the pan effect changes.

Effect Value

Let's return line 38 to the first `setEffect()` function call, which changed the volume of the drums on track 3:

```
38 setEffect(3, VOLUME, GAIN, 12.0)
```

Unit 2: Effects and Beats

2.1 Intro to Effects

What is an Effect?
How to use setEffect()

2.2 Intro to Beats

Beats, Meter, and Rhythm
Using Custom Beat Patterns and Strings

2.3 Saving Your Script in EarSketch and Uploading it to the Social Media Site

2.1 Intro to Effects

What is an effect?

(Return to Top)

In music production, a variety of effects can be added to a track in order to change the way a section of audio sounds. Effects are an important part of making electronic music. Electronics, from the simple transistor to the computer, have given musicians another tool to shape, process, manipulate, and change sound. Modern music production relies on effects to create mood, excitement, and musical style.

EarSketch supports a variety of effects that are common in music production. Some examples of these effects are listed below. For a complete list of the effects, **click here.**

Here is a reference sound without any effects applied to it. As we experiment with effects in this section, we will hear how the reference sound changes as the effects are applied to it.

Reference sound (the original sound with no effects):

Contents

- Welcome
- Unit 1: Getting Started
- Unit 2: Effects and Beats
- Unit 3: For Loops, User-Defined Functions, Debugging
- Unit 4: Lists, Effects in More Detail, Musical Form
- Unit 5: Randomness
- Unit 6: Sonification
- Unit 7: Teaching Computers
- Unit 8: Creating effects
- Unit 9: Recursion
- Reference
  - Recording
  - Creating Beats with makeBeat()
  - EarSketch API
  - EarSketch Sound Library
  - Every Effect Explained in Detail
  - Every Analysis Feature Explained in Detail
  - Python Reference
  - Copyright
- Additional Examples
  - Creating a "Fractalized Arrangement"
  - Effect Case Studies
- Teaching Resources
  - Teacher Materials
  - Social Media Teaching Resources
- Teacher Curriculum
  - Teacher Module 1: An Overview of Musical Concepts
  - Teacher Module 2: Rhythms
  - Teacher Module 3: Pattern and Variety
  - Teacher Module 4: Effects
  - Teacher Module 5: FAQ
- Old Curriculum

Unit 3: For Loops, User-Defined Functions, Debugging

3.1 For Loops
Fills

3.2 Conditionals

3.3 User-Defined Functions
How to Make a User-Defined Function
Return Values and Variable scoping

3.4 Debugging

3.1 For Loops

(Return to Top)

There are many programming situations where you may want to have a section of code execute several times. For instance, let's say that we want to have a single, one measure rhythm play on track 1 for 8 measures. At first, we might take this approach to the problem:

```
1  from earsketch import *
2  init()
3  setTempo(120)
4
5  beat = "0-00-00-0+++0+0+"
6  drum = ELECTRO_DRUM_MAIN_BEAT_008
7
8  makeBeat(drum, 1, 1, beat)
9  makeBeat(drum, 1, 2, beat)
10 makeBeat(drum, 1, 3, beat)
11 makeBeat(drum, 1, 4, beat)
12 makeBeat(drum, 1, 5, beat)
13 makeBeat(drum, 1, 6, beat)
14 makeBeat(drum, 1, 7, beat)
15 makeBeat(drum, 1, 8, beat)
16
17 finish()
```

This will yield the desired results. However, there's a sign that there is a better way of doing things here... the code is very repetitive and difficult to read. Because programmers are always looking for simpler and clearer ways of representing common situations, there are techniques to accomplish the same thing while typing less. In this case, it would be much simpler to use a **`for` loop**. The **`for` loop** structure makes the code appear much more readable and clearly indicates the logic. A `for` loop works by being given a certain range of values through which to operate. This range is described by the **`range()`** function.

Exercise

Change your script to match the latest version of Buzzjam "Lite", as shown in the following code:

```
#
#
# script_name: Buzzjam Lite
#
# author:
#
# description:
#
#
#
#setup section
from earsketch import *

init()
setTempo(120)

#music section
synth1 = TECHNO_SYNTHPLUCK_001
synth2 = TECHNO_SYNTHPLUCK_002
drums1 = TECHNO_LOOP_PART_002
drums2 = TECHNO_LOOP_PART_006

#melody
fitMedia(synth1, 1, 1, 2)
fitMedia(synth2, 2, 2, 3)
fitMedia(synth1, 1, 3, 4)
fitMedia(synth2, 2, 4, 5)
fitMedia(synth1, 1, 5, 6)
fitMedia(synth2, 2, 6, 7)
fitMedia(synth1, 1, 7, 8)
fitMedia(synth2, 2, 8, 9)
#drums
fitMedia(drums1, 3, 3, 9)
fitMedia(drums2, 4, 5, 9)

#effects
setEffect(3, VOLUME, GAIN, 12.0)

#finish section
finish()
```

-

```
#
#
# script_name:
#
# author:
#
# description:
#
#
#
#setup section
from earsketch import *

init()
setTempo(120)

#music section
synth1 = TECHNO_SYNTHPLUCK_001
synth2 = TECHNO_SYNTHPLUCK_002
introBeat = TECHNO_LOOP_PART_002
drums = TECHNO_LOOP_PART_006

#beats
introBeatString1 = "0-------0-------"
makeBeat(introBeat, 3, 4, introBeatString1)
makeBeat(introBeat, 3, 5, introBeatString1)
makeBeat(introBeat, 3, 6, introBeatString1)
makeBeat(introBeat, 3, 7, introBeatString1)
makeBeat(introBeat, 3, 8, introBeatString1)

#melody
fitMedia(synth1, 1, 1, 2)
fitMedia(synth2, 2, 2, 3)
fitMedia(synth1, 1, 3, 4)
fitMedia(synth2, 2, 4, 5)
fitMedia(synth1, 1, 5, 6)
fitMedia(synth2, 2, 6, 7)
fitMedia(synth1, 1, 7, 8)
fitMedia(synth2, 2, 8, 9)

#drums
fitMedia(drums, 4, 5, 9)

#effects
setEffect(3, VOLUME, GAIN, 12.0)

#finish section
finish()
```

Please Play:

Darryl McCune Introduces EarSketch Video

Guthman/Moog Student Design Challenge

February 15, 2015

Michael J. Adams
President & CEO
Moog Music Inc.

WERKSTATT-01

QUICK START

To Learn More About Your Werkstatt Analog Synthesizer Download The Manual At: Moogmusic.com/werkstatt

VCO

A single Voltage Controlled Oscillator with two selectable waveforms is Werkstatt's primary source of sound.

FREQ - The Frequency knob sets the pitch of the VCO. Its range is 9+ octaves.

WAVE - Saw and Pulse waveforms are available on the Werkstatt. Each carries a distinct harmonic content, or timbre. When set to SAW the PWM knob is disabled.

PWM - The Pulse Width Modulation knob can be used to determine the width of the Pulse wave, thereby changing the timbre of the wave.

VCO MOD

Oscillator Modulation is a way of creating everything from subtle pitch vibrato, and chorus-like sounds to metallic and harsh effects.

SOURCE - The Source switch determines which Modulation Source will be applied to the Oscillator. ***EG:*** The Attack, Sustain, and Decay characteristics of the Envelope Generator will be used as the VCO modulation source. ***LFO:*** The Low Frequency Oscillator will be used as the VCO modulation source.

AMOUNT - The Amount knob is used to control the depth of the modulation.

DEST - The Destination switch determines which control will be affected by the chosen Modulation Source.With this switch set to PWM, the PWM knob is disabled. ***PWM:*** The Modulation source is used to continually vary the width of the Pulse wave. The VCO WAVE switch must be set to Pulse. ***FREQ:*** The Modulation source is used to continually vary the pitch of the Oscillator.

LFO

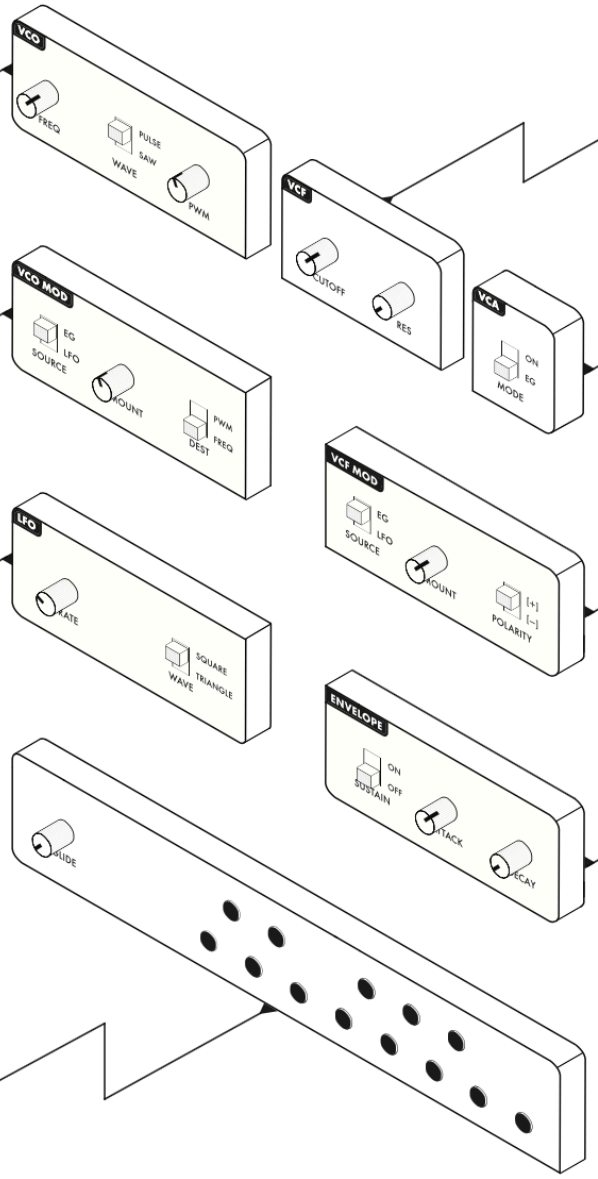
The Low Frequency Oscillator creates a cyclical modulation source that can be used to add a repeating change to any modulation destination.

RATE - Controls the speed, or frequency, of the LFO. The LED flashes once for every wave cycle, displaying the LFO speed.

WAVE - This switch allows you to select the Waveform of the LFO. ***SQUARE:*** The LFO will alternate directly between two distinct values representing the upper and lower limits of the wave. ***TRIANGLE:*** Creates a continuously changing value that sweeps between the upper and lower limits of the wave.

KBD - The keyboard features one-octave of round buttons as opposed to traditional keys, but the layout is the same. If more than one note is played at a time, the lowest note will be played.

GLIDE - Determines the time it takes to make a smooth pitch transition from one note to another note.



VCF

The Voltage Controlled Filter is a classic 24dB per octave Moog Ladder filter, which shapes sound by attenuating and/or emphasizing certain harmonic elements.

CUTOFF - The Cutoff knob specifies the frequency at which the Filter begins to attenuate sound.

RES - The Resonance knob determines the amount of harmonic emphasis applied to the filter cutoff frequency.

VCA

The Voltage Controlled Amplifier determines the output level of the Werkstatt. Its source is determined by the Mode switch.

MODE - The VCA can function in one of two ways, selected by the Mode switch. ***ON*** - The audio signal of the Werkstatt will output, or drone continuously. ***EG*** - The audio output level is controlled by the Attack, Sustain, and Decay characteristics of the Envelope Generator Module.

VCF MOD

Filter Modulation changes the value of the Filter's Cutoff Frequency.

SOURCE - The Source switch determines which Modulation Source will be applied to the filter cutoff. ***EG:*** The Attack, Sustain, and Decay characteristics of the Envelope Generator will be used as the VCF modulation source. ***LFO:*** The Low Frequency Oscillator will be used as the VCF modulation source.

AMOUNT - The Amount knob is used to control the depth of the modulation.

POLARITY- When EG is the modulation source, its Polarity can be reversed. This allows the envelope Attack to lower the Filter Cutoff Frequency instead of boosting it. **[+]:** Normal Polarity **[-]:** Reverse Polarity.

ENVELOPE

Each time a key is pressed, the Envelope Generator produces control voltages that allow you to change the value of certain parameters over time.

SUSTAIN - Determines how the Sustain segment of the Envelope Generator will work when a note is held. ***ON:*** The Envelope will continue to sustain as long as the keyboard key being held. This results in an organ-like behavior. ***OFF:*** The Envelope will advance directly from the Attack stage to the Decay stage. This results in a plucked sound.

ATTACK - Determines the time it takes for the Envelope Generator to reach its maximum level after a key is pressed.

DECAY - Determines the time it takes for the Envelope Generator to reach its lowest level after the key is released, or after the attack segment is complete.

PATCHABLE HEADER - This allows control signals generated by the Werkstatt to be easily routed to the control inputs of the VCO, VCF, VCA, and LFO. The patch wires included with the Werkstatt are designed to make the most of these patch points.

WERKSTATT-01

ASSEMBLY INSTRUCTIONS

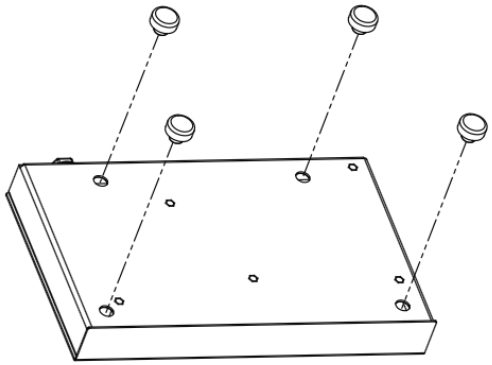
PARTS LIST

A) Werkstatt Ø1 Printed Circuit Board (PCB)..... x 1
B) Bottom Chassis (metal)..... x 1
C) Top Panel with silkscreened printing(metal)..... x 1
D) 12 Volt DC power adapter..... x 1
E) Hardware Kit containing:
- a. Keyboard button caps..... x 13
- b. 1/4" Sheet metal screws (Black)..... x 4
- c. 1/4" Pan head machine screws (Silver)..... x 5
- d. Black nylon washer..... x 1
- e. Black nylon hex nut..... X 1
- f. Rubber feet..... x 4
- g. Patchable Header cables..... x 5

F) Serial Number label..... x 1

1. ATTACHING THE FEET

Attach the four rubber feet by pushing them into the holes of the Bottom Chassis and gently twisting.

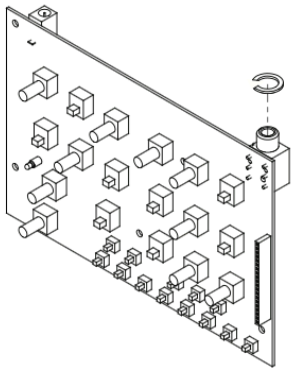


2. TRIMMING THE NYLON WASHER

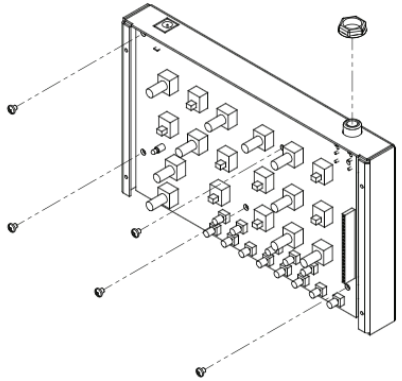
The black nylon washer is used to isolate the audio jack from the inside of the Bottom Chassis. In order to slide the washer onto the audio jack, you will need to cut a 1/16" section from the washer.

3. INSTALLING THE ELECTRONICS

Carefully remove the Printed Circuit Board from the protective sleeve. Take the washer you prepared in the previous step and slide it onto the audio output jack, lining up the missing section with the metal tab on the jack.



Next, place the Printed Circuit Board into the Bottom Chassis. Slide the audio jack through the jack hole, and be sure the five mounting holes in the Printed Circuit Board line up with the standoffs on the Bottom Chassis. Using the silver pan head machine screws (x5), loosely attach the Printed Circuit Board to the chassis. Take a moment to make sure everything is lined up neatly and correctly—including the power input jack—and then go ahead and tighten the screws.



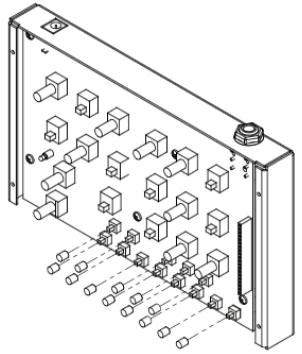
Finally, place the black nylon hex nut on the audio jack, and hand-tighten the nut to hold the jack securely to the Bottom Chassis.

4. POWER/LED TEST & KEYBOARD

Using only the included power supply, connect the barrel end to the power supply input of your Werkstatt-Ø1; connect the other end to an AC wall outlet (100-240 Volts AC / 50-60 Hz). At this point, the red LED on the front panel (LFO) should begin blinking. If it does, you're ready to move on.

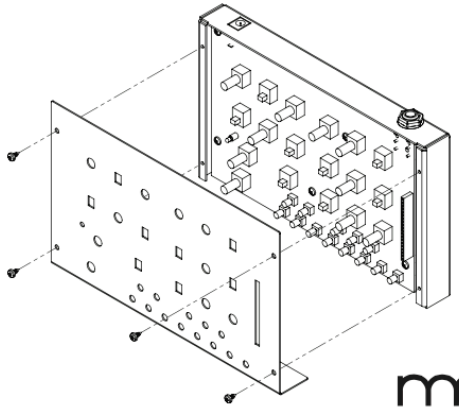
Unplug the power supply from the Werkstatt before proceeding.

Place the 13 button caps on the keyboard buttons.

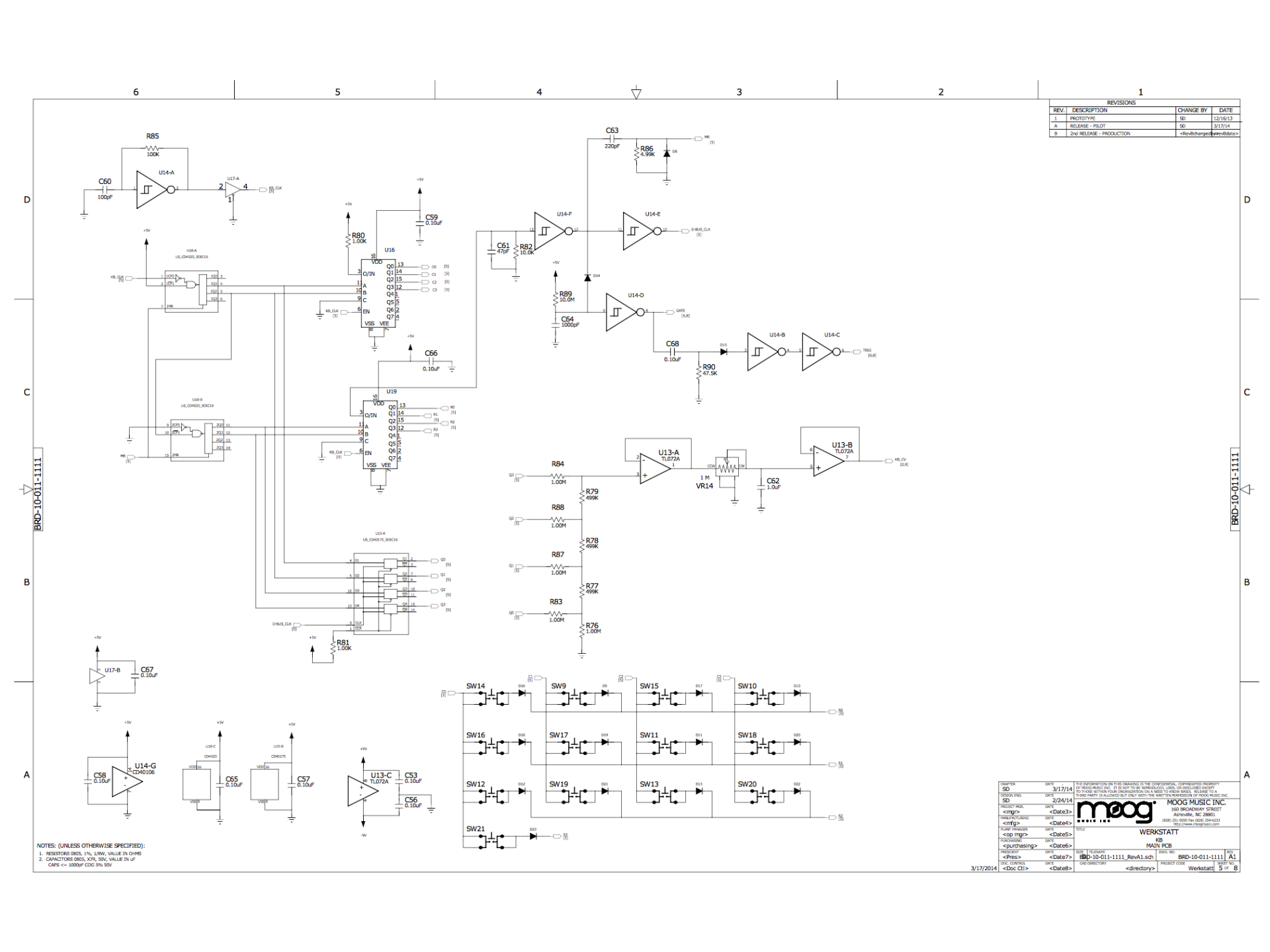


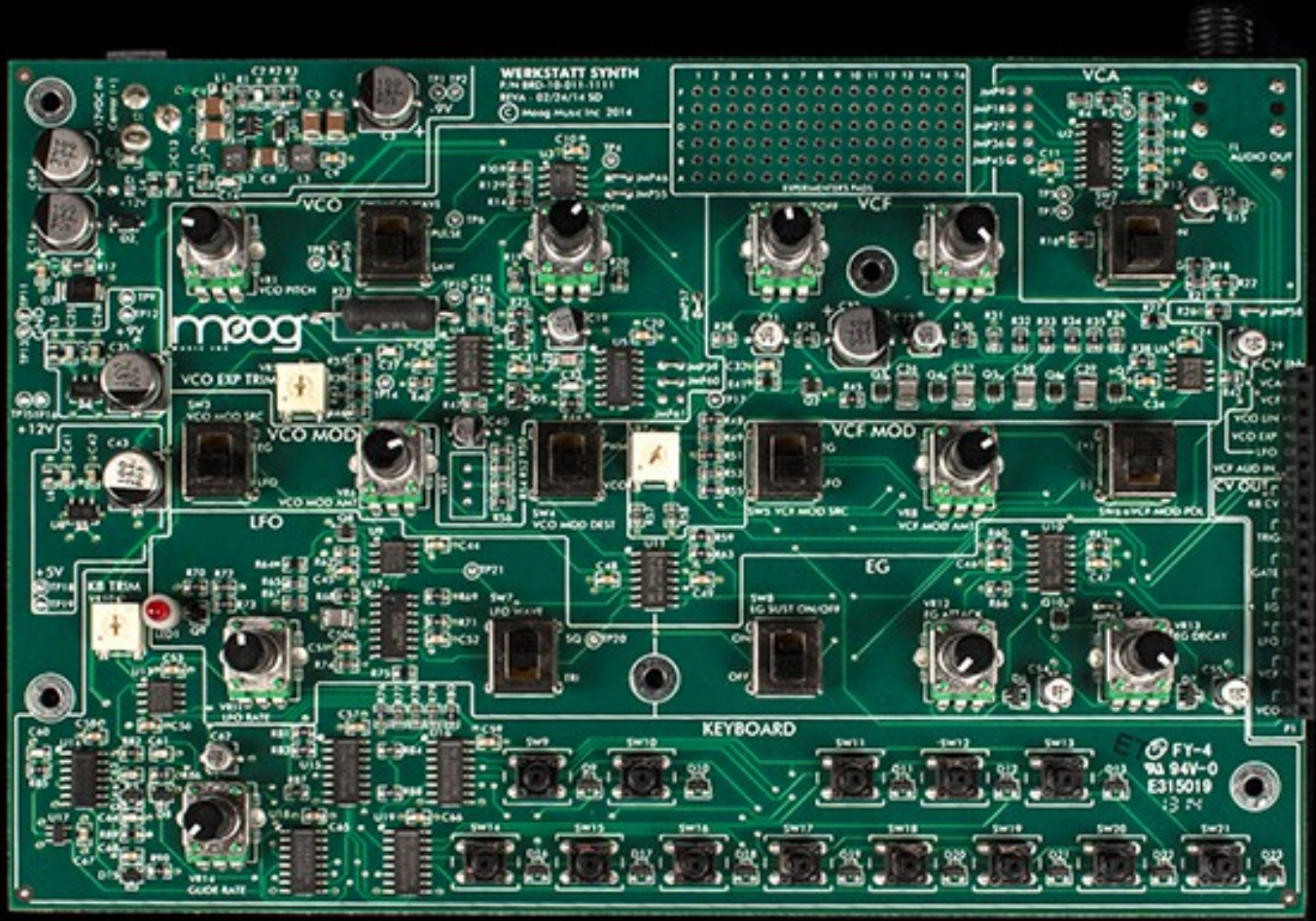
5. ADDING THE TOP PANEL

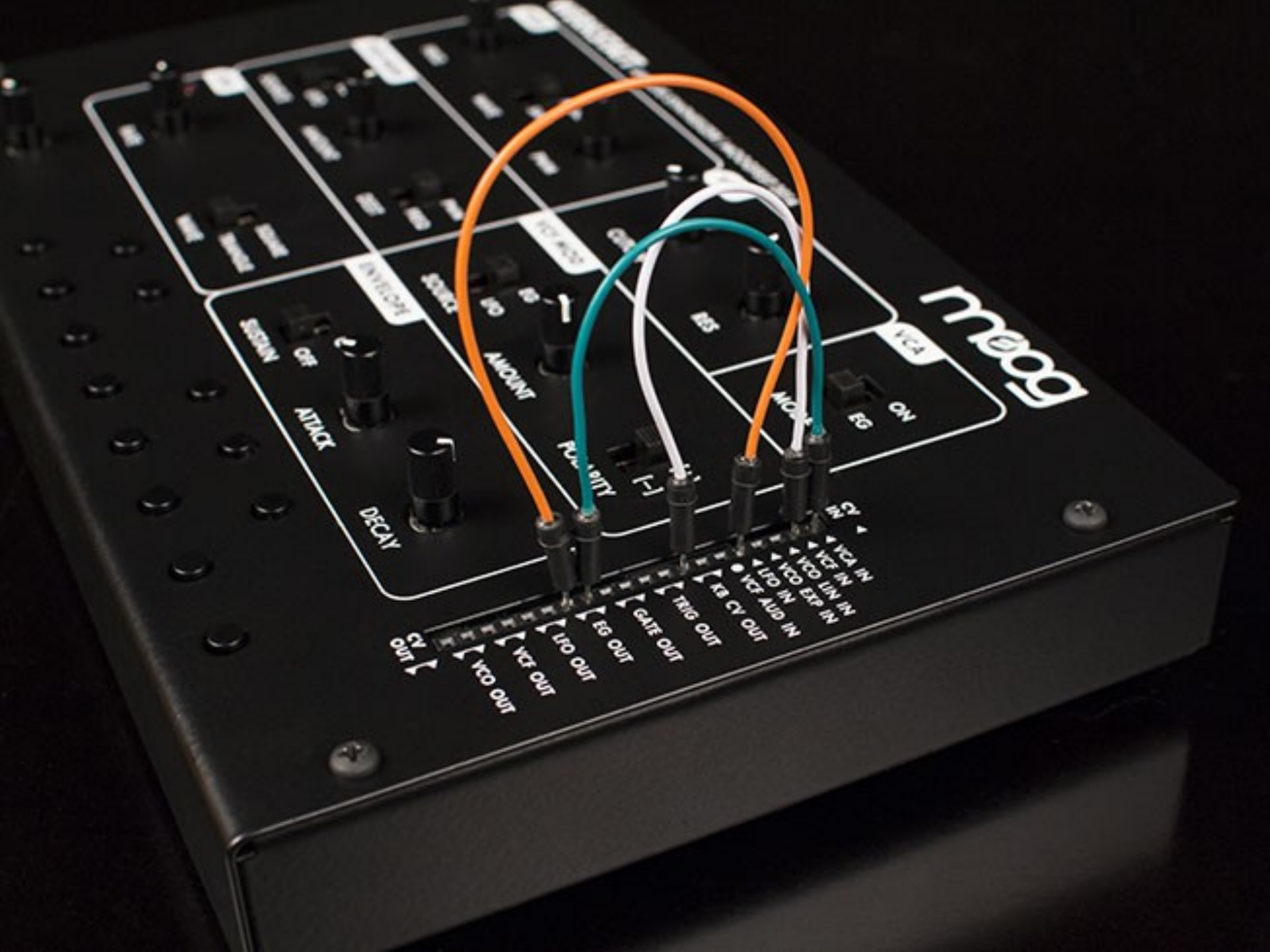
Next, place the Top Panel in position so that all of the knobs, switches, and keyboard buttons pass through the corresponding holes. Secure the Top Panel using the black sheet metal screws. The screw holes in the Top Panel should line up with the screw holes in the Bottom Chassis. *NOTE: Tightening these screws may require a little extra effort the first time.*

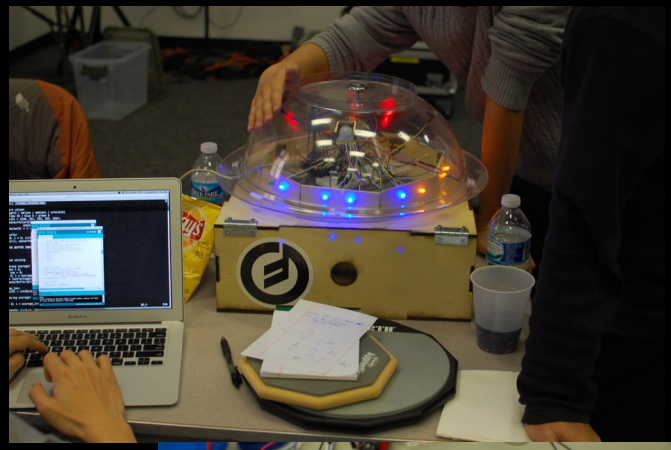


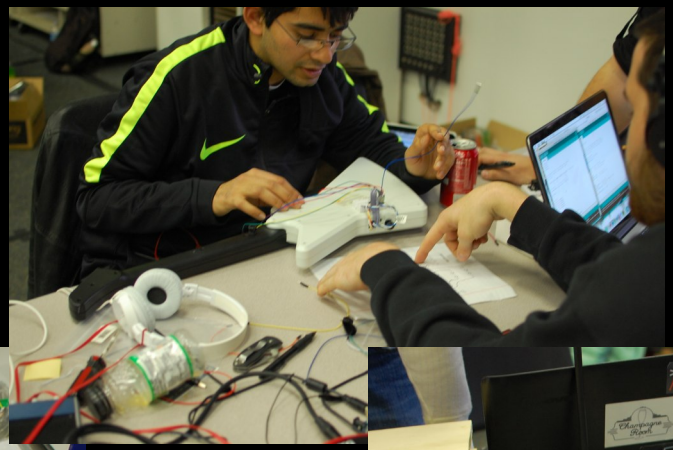
moog

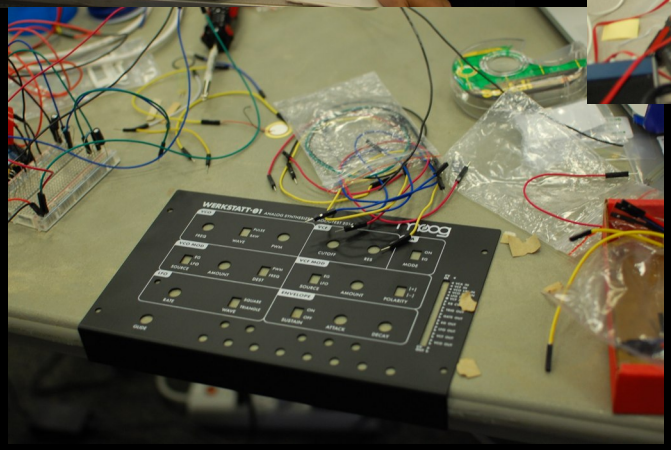

REVISIONS
REV. | DESCRIPTION | CHANGE BY | DATE
1 | PROTOTYPE | SD | 12/16/13
A | RELEASE - PILOT | SD | 3/17/14
B | 2nd RELEASE - PRODUCTION | <RevBchangedby> | <RevBdate>
BRD-10-011-1111
R85 100K
C60 100pF
U14-A
U17-A
KB_CLK [5]
C59 0.10uF
R80 1.00K
U16
C63 220pF
R86 4.99K
D6
MR [5]
U14-F
U14-E
C61 47pF
R82 10.0K
D-BUS_CLK [5]
D14
R89 10.0M
C64 1000pF
U14-D
GATE [5,8]
C68 0.10uF
D15
R90 47.5K
U14-B
U14-C
TRIG [6,8]
C66 0.10uF
U19
R84 1.00M
R79 499K
R88 1.00M
R78 499K
R87 1.00M
R77 499K
R83 1.00M
R76 1.00M
U13-A TL072A
1 M VR14
C62 1.0uF
U13-B TL072A
KB_CV [2,8]
R81 1.00K
U17-B
C67 0.10uF
U14-G CD40106
C58 0.10uF
C65 0.10uF
C57 0.10uF
U13-C TL072A
C53 0.10uF
C56 0.10uF
SW14
SW9
SW15
SW10
SW16
SW17
SW11
SW18
SW12
SW19
SW13
SW20
SW21
NOTES: (UNLESS OTHERWISE SPECIFIED):
1. RESISTORS 0805, 1%, 1/8W, VALUE IN OHMS
2. CAPACITORS 0805, X7R, 50V, VALUE IN uF
CAPS <= 1000pF C0G 5% 50V
DRAFTER SD 3/17/14
DESIGN ENG. SD 2/24/14
PROJECT MGR. <mgr> <Date3>
MANUFACTURING <mfg> <Date4>
PLANT MANAGER <op mgr> <Date5>
PURCHASING <purchasing> <Date6>
PRESIDENT <Pres> <Date7>
DOC. CONTROL <Doc Ctl> <Date8>
THE INFORMATION ON THIS DRAWING IS THE CONFIDENTIAL, COPYRIGHTED PROPERTY OF MOOG MUSIC INC. IT IS NOT TO BE REPRODUCED, USED, OR DISCLOSED EXCEPT AS AUTHORIZED IN WRITING BY MOOG MUSIC INC. IT IS NOT TO BE GIVEN TO A THIRD PARTY. IS ALLOWED BUT ONLY WITH THE WRITTEN PERMISSION OF MOOG MUSIC INC.
MOOG MUSIC INC.
160 BROADWAY STREET
Asheville, NC 28801
(828) 251-0090 Fax (828) 254-6233
http://www.moogmusic.com
TITLE
WERKSTATT
KB
MAIN PCB
SIZE B
FILENAME BRD-10-011-1111_RevA1.sch
DWG. NO. BRD-10-011-1111
REV. A1
CAD DIRECTORY <directory>
PROJECT CODE Werkstatt
SHEET NO. 5 OF 8
3/17/2014

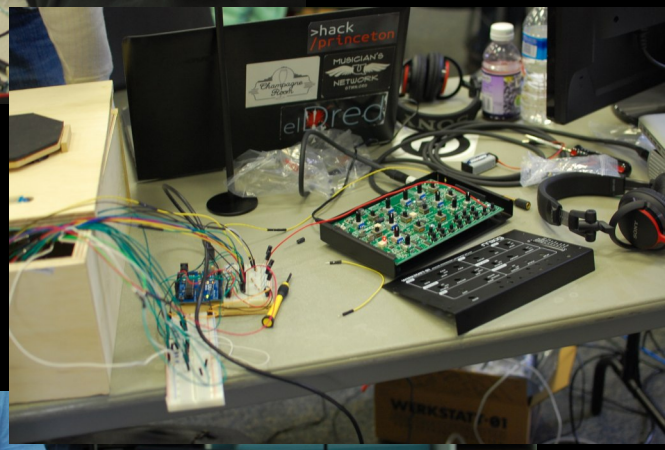

WERKSTATT SYNTH
VCO
VCF
VCA
VCO MOD
VCF MOD
LFO
EG
KEYBOARD
moog

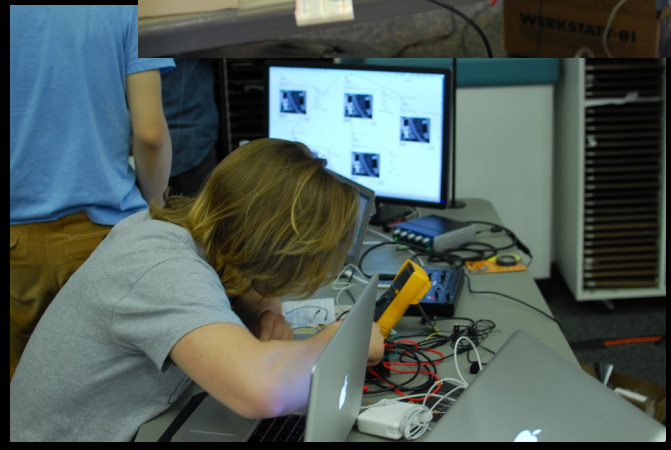

moog
SUSTAIN
OFF
ATTACK
DECAY
SOURCE
AMOUNT
POLARITY
VCA
ON
EG
CV OUT
VCO OUT
VCF OUT
LFO OUT
EG OUT
GATE OUT
TRIG OUT
KB CV OUT
VCF AUD IN
LFO IN
VCO EXP IN
VCO LIN IN
VCF IN
VCA IN
CV IN





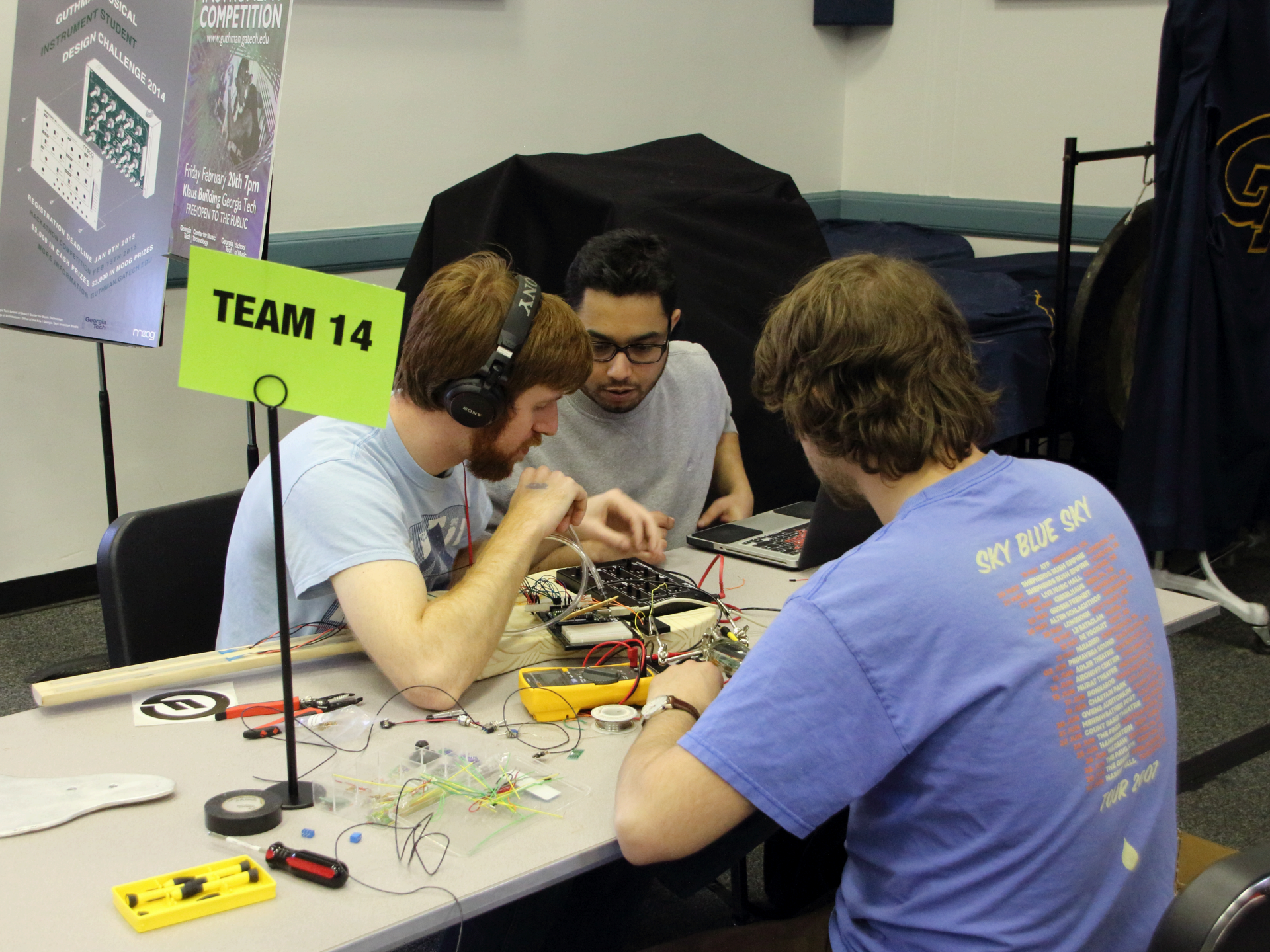

WERKSTATT-01
moog


shack
princeton
MUSICIAN'S
NETWORK

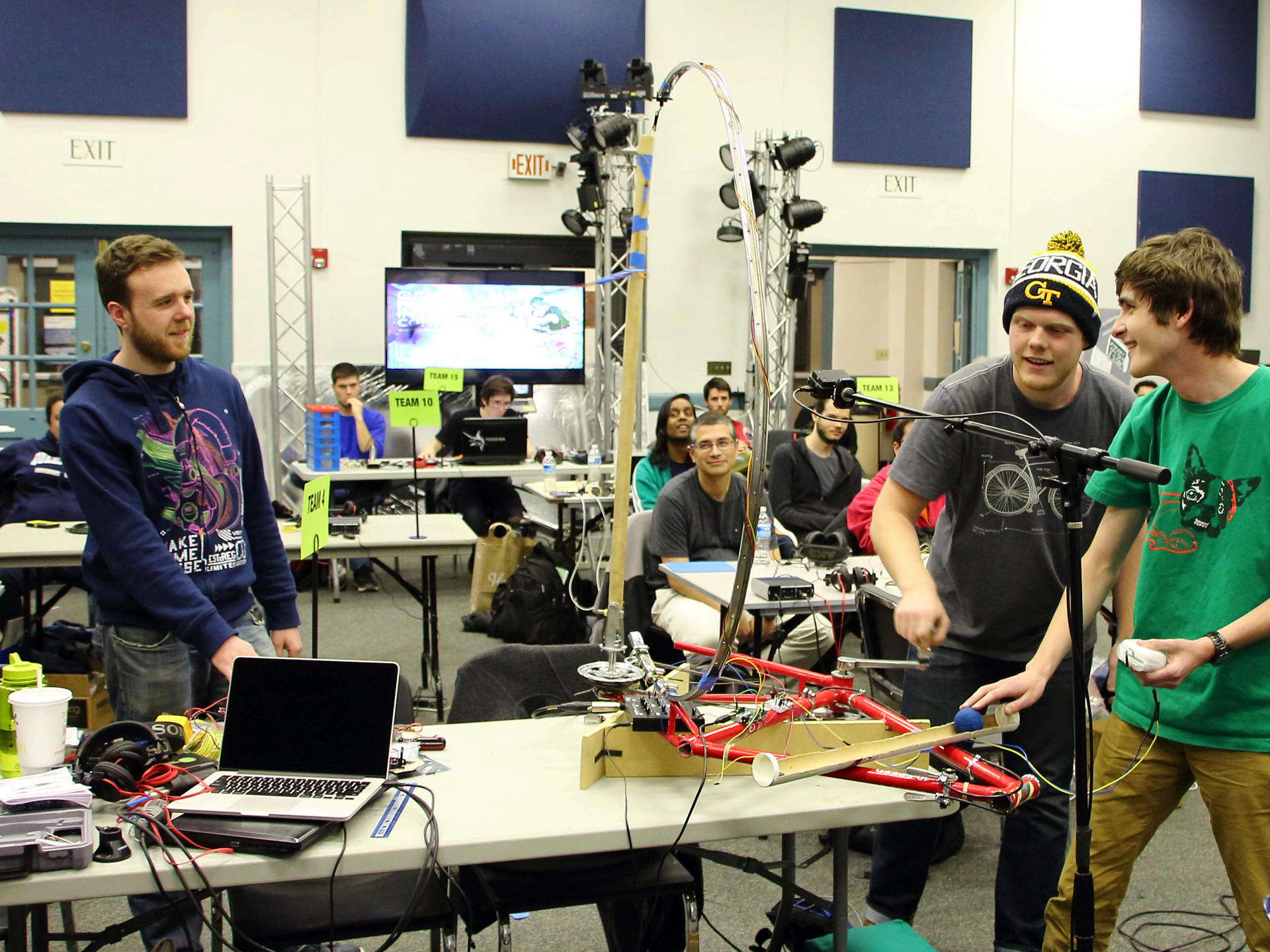



TEAM 12


Georgia Tech
School of Music


INSTRUMENT STUDENT
DESIGN CHALLENGE 2014
COMPETITION
www.guthman.gatech.edu
Friday February 20th 7pm
Klaus Building Georgia Tech
FREE/OPEN TO THE PUBLIC
TEAM 14
SKY BLUE SKY


TEAM 9
New Orleans


EXIT
EXIT
EXIT
TEAM 15
TEAM 10
TEAM 13
TEAM 4
GEORGIA
GT


TEAM 2


DEER PARK


YAMAHA


Georgia Tech
School of Music
College of Architecture

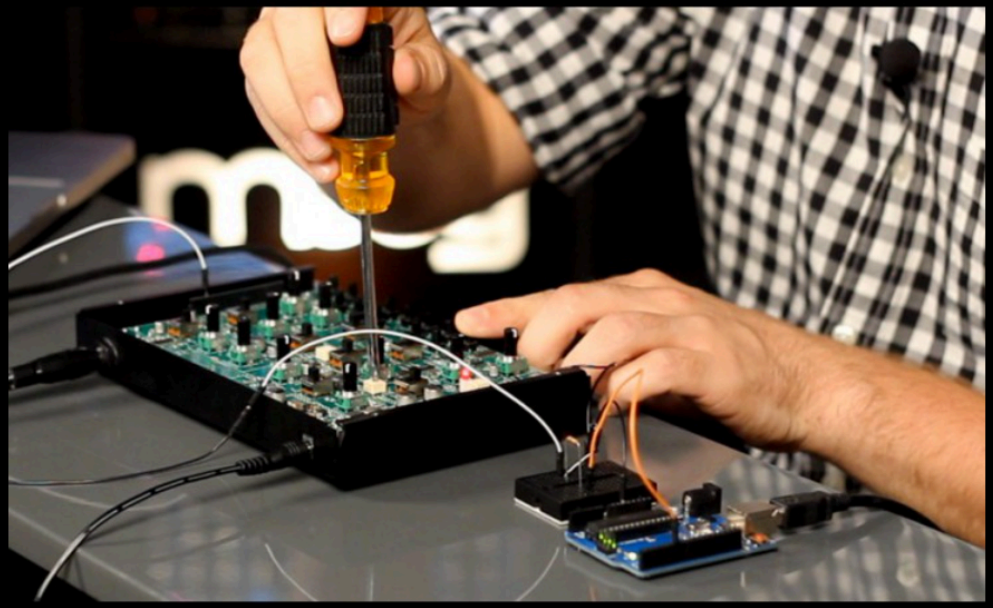

GUTHMAN
MUSICAL
INSTRUM
COMPET
BOSS

Moog Werkstatt Curriculum

- Developed by Chris Howe, MSMT student at Georgia Tech.
- Expanded in his Masters Thesis – ANALOG SYNYTHESIZERS IN THE CLASSROOM: How creative play, musical composition, and interest driven learning can enhance STEM standard mastery.
- Research includes the development of a STEAM curriculum revolving around the Moog Werkstatt synthesizer. The curriculum aims to meet the standards presented by either Common Core or Next Generation Science Standards.
- Currently, modules are available at the following links:

  http://chrisdavidhowe.drupalgardens.com
  http://www.moogmusic.com/products/werkstatt/werkstatt-workshop
  http://www.werkstattworkshop.com

WERKSTATT WORKSHOP



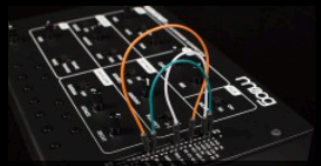
INFO

WerkstattWorkshop.com (beta) is Moog Music's interactive creative learning portal containingproject ideas, mod tutorials, parts lists, educational lesson plans, 3D printer files, and everything else involved with learning and modifying your Werkstatt. Moog Music encourages all that are interested to share their Werkstatt mods with the community by submitting them at werkstattworkshop.com.

This site also features our current research and lessons surrounding the use of subtractive synthesizers in STEM and STEAM high-school classrooms. Lessons in these studies are focused on project based learning opportunities, and encouraging a creative deployment of practical skills in these fields. All lessons and associated materials are open source and available online for educators across multiple disciplines.

VISIT THE SITE

WerkstattWorkshop.com (beta) is Moog Music's interactive creative learning portal containingproject ideas, mod tutorials, parts lists, educational lesson plans, 3D printer files, and everything else involved with learning and modifying your Werkstatt.



WERKSTATT-01 MOOGFEST 2014 KIT

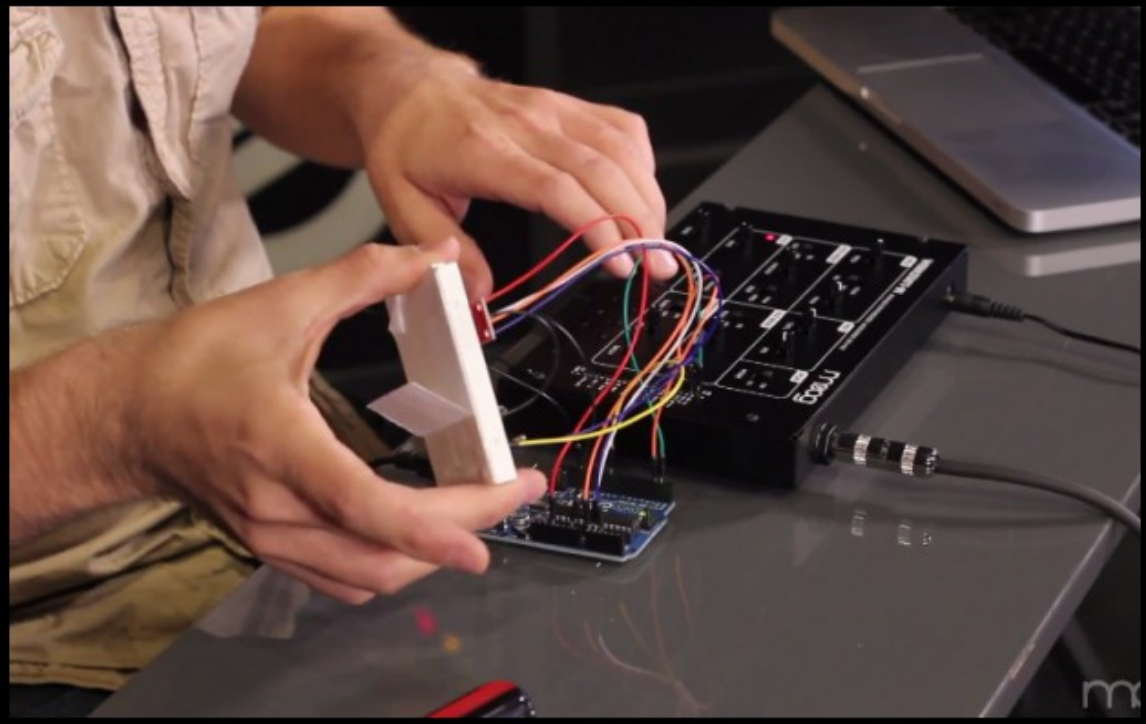
WERKSTATT CV EXPANDER

WERKSTATT-01 CABLE SET

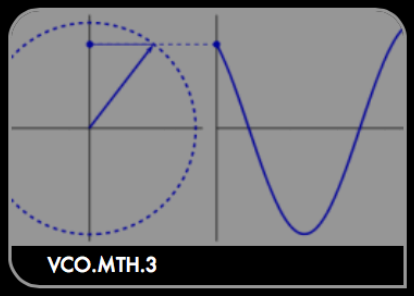
This site also features our current research and lessons surrounding the use of subtractive synthesizers in STEM and STEAM high-school classrooms. Lessons in these studies are focused on project based learning opportunities, and encouraging a creative deployment of practical skills in these fields. All lessons and associated materials are open source and available online for educators across multiple disciplines.

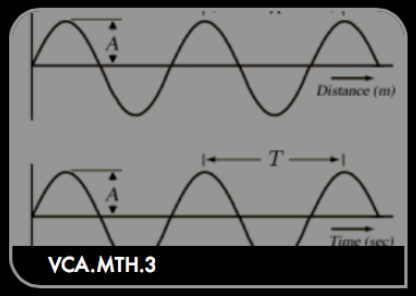
Current Projects:

- Pitchbend
- Arpeggiator/Sequencer
- Noise Generator
- Push Button Vibrato
- LFO Rate Quantizer
- Dual Oscillator
- Volume Knob
- 555 IC - 2nd Oscillator
- Photocell

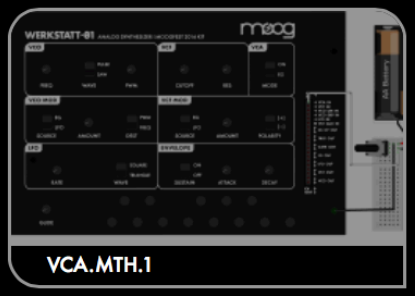


Mathematics

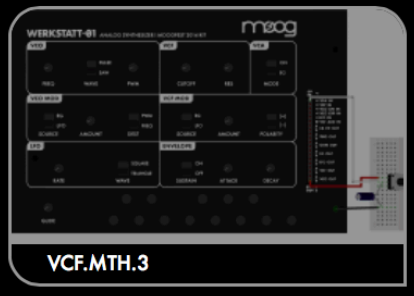

VCO.MTH.3



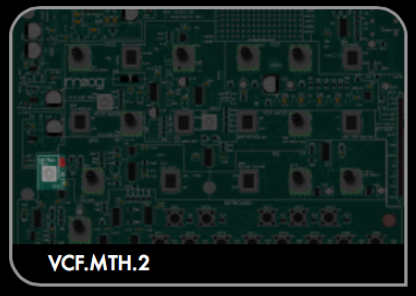
VCA.MTH.3



VCA.MTH.1



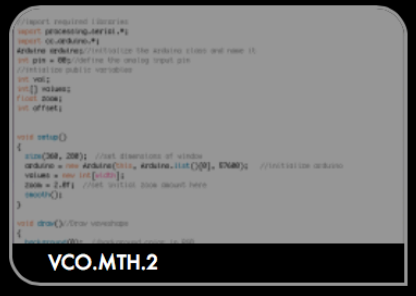
VCF.MTH.3



VCF.MTH.2


VCF.MTH.1


VCO.MTH.4


VCO.MTH.2

VCO.MTH.1

Please Play:

WERKSTATT-01 Push Button Vibrato
Moog Music Inc Video

VCO.TECH.3

Lesson Summary

Construct a similar sounding VCO setting and evaluate the scientific principles behind that specific timbre.

Skills

Vocabulary

Timbre - Also known as tone color or tone quality. Timbre the quality of a musical note or sound or tone that distinguishes different types of sound. The physical characteristics of sound that determine the perception of timbre include a sounds frequency spectrum and envelope.

Fundamental frequency - the harmonic component of a complex wave that has the lowest frequency and commonly the greatest amplitude. In musical terms the principal tone produced.

Harmonics – Higher frequencies of a complex waveform that are multiples of the fundamental frequency. For example if the fundamental is 200Hz, the second harmonic is 400Hz and third harmonic is 600Hz.

Exercise

Recreate the bass setting from Stevie Wonder's "Boogie on Reggae Woman".

Materials

Werkstatt

Vocabulary

Timbre - Also known as tone color or tone quality. Timbre the quality of a musical note or sound or tone that distinguishes different types of sound. The physical characteristics of sound that determine the perception of timbre include a sounds frequency spectrum and envelope.

Fundamental frequency - the harmonic component of a complex wave that has the lowest frequency and commonly the greatest amplitude. In musical terms the principal tone produced.

Harmonics – Higher frequencies of a complex waveform that are multiples of the fundamental frequency. For example if the fundamental is 200Hz, the second harmonic is 400Hz and third harmonic is 600Hz.

Exercise

Recreate the bass setting from Stevie Wonder's "Boogie on Reggae Woman".

Materials

Werkstatt

Practice

Find a song that uses electronic musical synthesis and do your best to emulate a sound from that song. Explain what processes you used in the Werkstatt, and how the sound was manipulated.

Standards

NGSS.ETS1.B

Subject

Engineering Technology

Unit

VCO

Georgia Tech Center for Music Technology

Music Technology Outreach

- Comments?
- Questions?
- Concerns?
- Rants?

TeknoDida 2015
17.–19.4.2015

TeknoDida 2015 | Perjantai | Lauantai | Sunnuntai

Frank Clark
Chair and Professor
School of Music
Georgia Institute of Technology
fclark@music.gatech.edu

MUSIC TECHNOLOGY OUTREACH