What is this?

Congratulations for acquiring Mixtape, the desktop music box synthesizer. Hopefully, you will be able to start exploring the world of electronic music with this kit and manual. For starters, what is Mixtape?

- It is a 4 Voice Electronic Synthesizer
- 4 bit independent VCA with Attack, Decay, Sustain, and Release (ADSR)
- 512kbit memory for storing music, plus demos that are always available
- 3 buttons to quickly load and play your music into memory
- Visual representation of Music and progress bar with LEDs
- Headphone jack with gain and volume controls
- USB powered, with USB Serial connection at 115200 baud
- Command line interface for entering music and controlling the synthesizer
- Musescore, MIDI, MusicXML, and other formats can be converted with Mixtape scripts or create your own with Humdrum-Tools and other libraries
- Bootloaded Firmware – Your feedback could result in firmware enhancements that can be distributed over the internet!

In your kit, you should have received the following: (3D printed part colors may vary)

- 1 Pre-assembled PCB board
- 2 Acrylic sheets, pre-cut/routed
- 4 3D printed case corners
- 8 Self-tapping screws (#4 x ½” wood screws)
- 3 3D printed switch caps
- 2 3D printed potentiometer knobs
- 1 USB-B to USB-A cable
- 1 Pair of earbuds

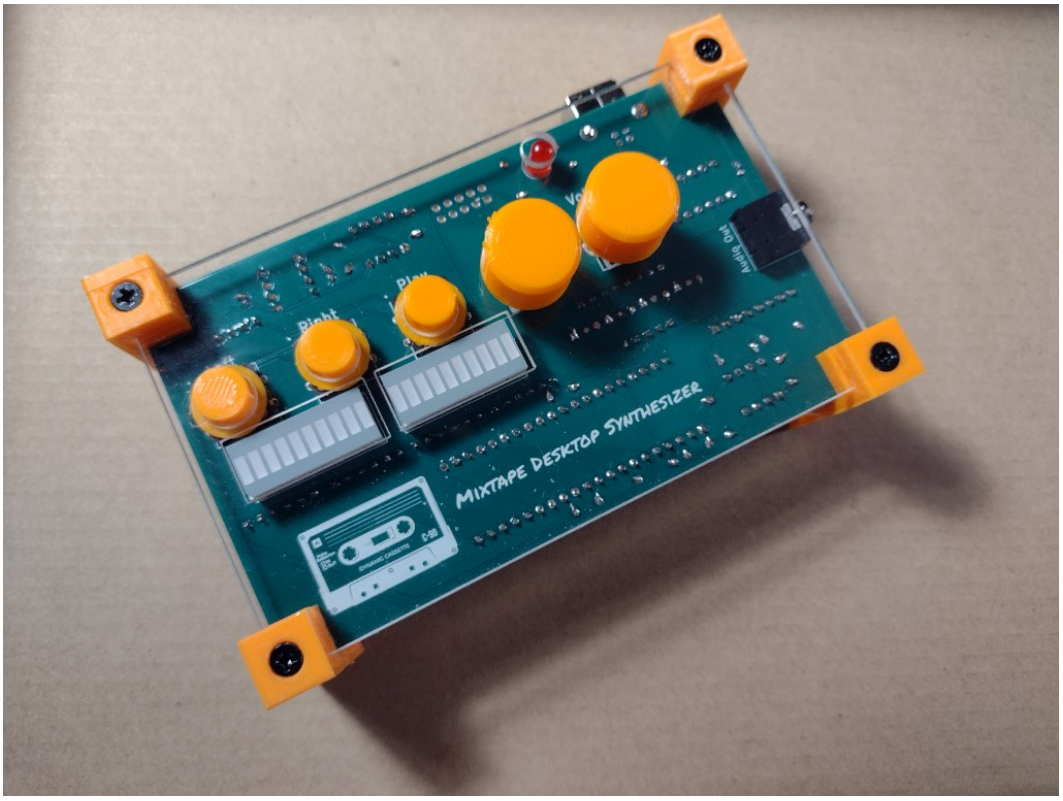
Some required extras:

- 1 Phillips screwdriver
- Computer (Linux, Mac, Windows)
- Serial Terminal Software(YAT or similar for Windows, CuteCom or Mo Serial for Linux, CoolTerm for Macintosh)

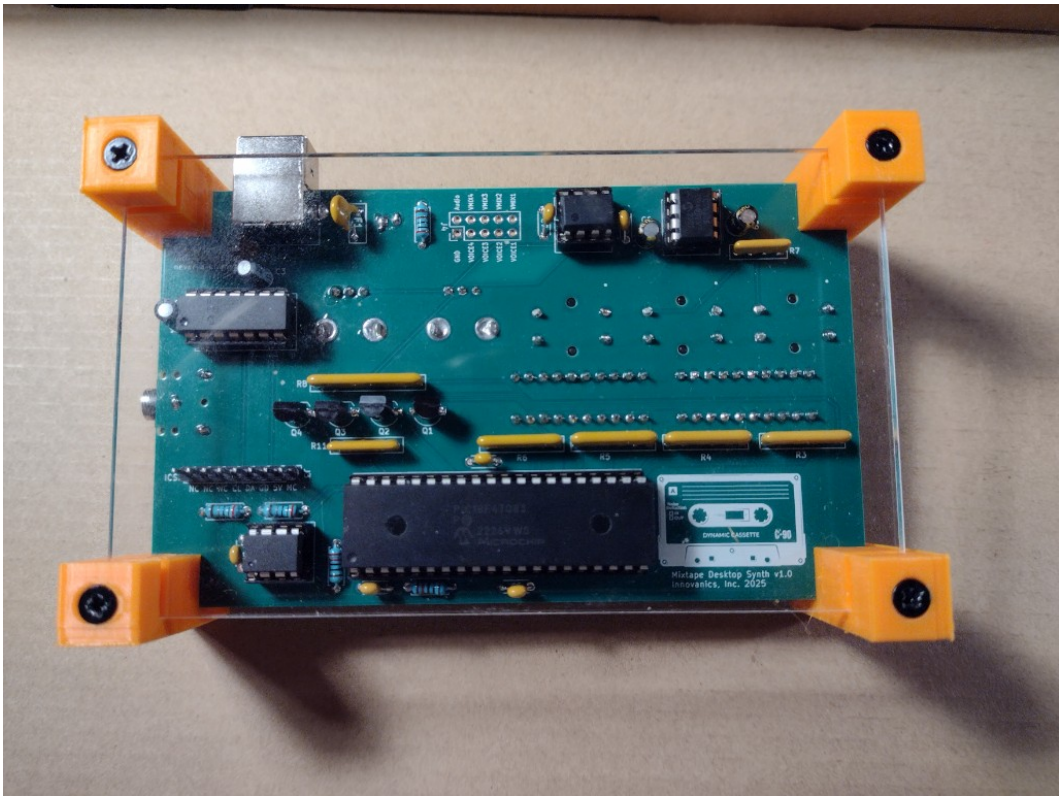
Some assembly is required, continue on...

I want to play! How do I set this up?

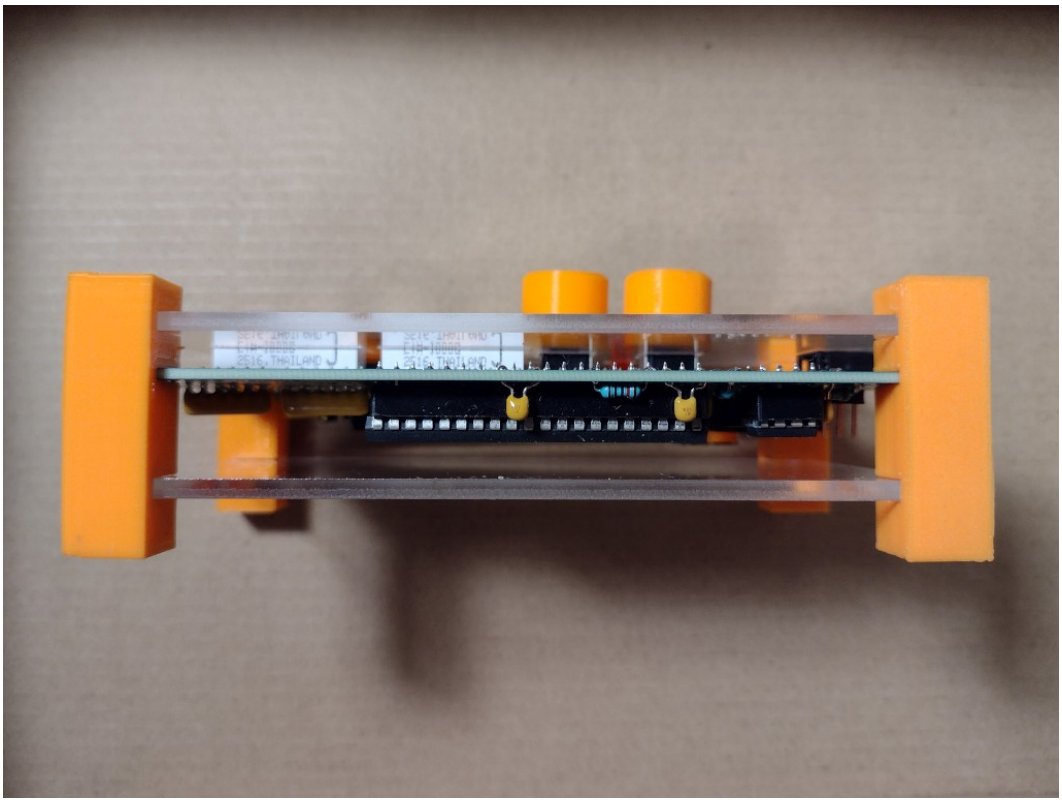
The case consists of 2 clear acrylic sheets that sandwich the printed circuit board (PCB) and button switches. All three layers fit into the 3D printed corners. When you are finished assembling Mixtape, it should look like the pictures below. Note that the screws are driven through the top and bottom acrylic sheets, and that the PCB is held in place with limited movement because it is the same size as the sheets. Note the orientation of the switches and cutouts on the top.


MIXTAPE DESKTOP SYNTHESIZER

Top View


Mixtape Desktop Synth v1.0
Innovanics, Inc. 2025

Bottom View



Side View

1. Remove the brown protective paper adhesive covering from the 2 acrylic sheets to reveal the clear acrylic underneath. To avoid fingerprints, handle the sheets by their edges. You can clean them with isopropyl alcohol if you leave fingerprints.
2. Take 1 3D printed corner and note that there is an up arrow, and is intended to sit on your desk pointing up. The spacing of the boards is different, and the cutouts won't fit unless the layers are correct. Insert one corner of the bottom clear acrylic sheet (without cutouts, only 4 drilled holes) in the lowest slot of the corner.
3. Use your Phillips screwdriver to gently drive one screw through the bottom, through the plastic sheet. Do not tighten too much, or the plastic will strip. It is suggested to half-tighten all screws at first, then do a final tightening at the end.

4. Take the 3 switch caps and put them on the 3 switches on the PCB. They might be a little bit loose, but will be held in place when the top cover is in place. Do NOT put on the volume knobs, yet.
5. Align the top acrylic sheet so that it fits on top of the switches and the printed circuit board. You may need to undo the top to reorient it if it is misaligned. The cutouts should fit the switches and LEDs.
6. Take the top clear acrylic cover and PCB and slide them into the corner. Screw the top in with 1 screw.
7. Align one more corner on an opposite side of the assembly, orienting it the same way as the first corner. Make sure all layers are inserted in the slots. Drive in the bottom and top screws.
8. Repeat for the remaining 2 corners.
9. Side the gain and volume knobs on top of the stems. They have a D connection that needs to be aligned with the knobs. Assembly is now Complete!!!!!!

Next, install the USB drivers for your computer.

Mixtape works with:

- Windows
- Macintosh
- Linux/BSD variants

Mixtape was developed on Linux, and has been tested on Windows 11 and Mac OS X 10.12.

If you are using Linux or Macintosh, the USB chip (Microchip's MCP2221A) does not require an additional driver. If you use Windows, please download the driver from Microchip Technology's website, and install.

https://www.microchip.com/en-us/product/mcp2221a

Instructions are included in the download. You should not need to download the chip utility, the chip has been pre-configured.

Plug the unit in with the included USB cable to your computer. Never power it from a USB charger! The unit is fuse protected, but too much current could still start a fire.

You will need to install a terminal program that can connect to a serial port. On Linux, try CuteCom, installable from the Software app, Flatpak, or SNAP. Windows users can use Putty, and Mac Users can try CoolTerm.

On Linux, the serial port will be similar to:

/dev/ttyACM0

If you have a mouse or device that uses the same driver, it might enumerate to /dev/ttyACM1 or /dev/ttyACM2. Once you plug in the device with USB into the

computer, type “sudo dmesg” and look at the message at the bottom. It should say that a USB connection has been established and list the serial port.

Windows users can find which COM port it is connected to by checking the Device manager. The device may also show up in the taskbar icon list.

On the Macintosh, try looking in the terminal, or checking “dmesg” output. It will come up automatically with CoolTerm.

Remember this serial port for later.

Let’s hear some music!

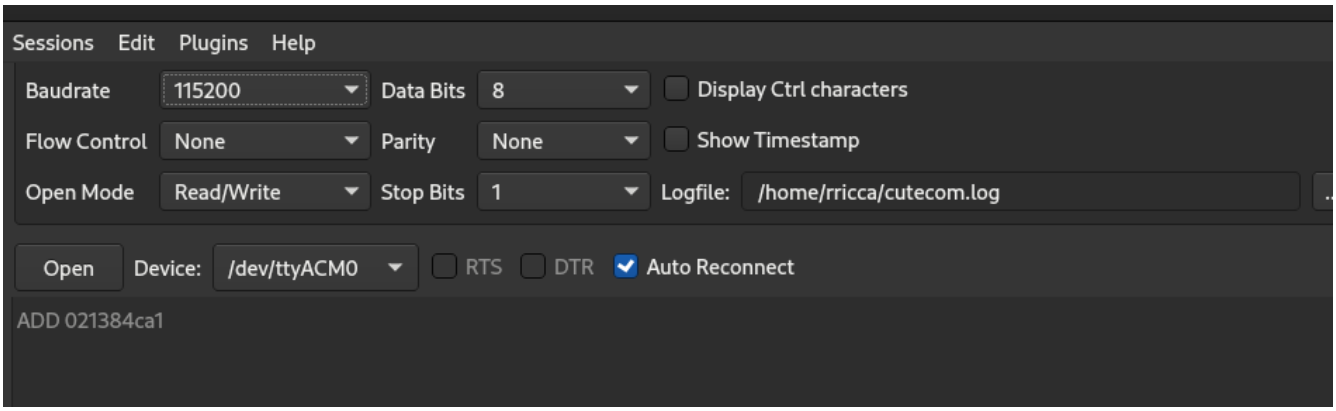
- When the device is plugged into your computer, plug in your headphones in the jack on the right of the unit. Your favorite headphones with a 3.5mm jack or the included headphones should work with Mixtape.
- There are a few songs pre-installed to demonstrate some of its capabilities.
- When you first start playing with Mixtape, you need to go into demo mode to listen to the samples.
- To get into DEMO mode, push in the button labeled “Right” (middle button) and hold it while you push the “Left” button (1st, leftmost button) and release the right then left. You can use a rocking motion “click middle, click left, release middle, release left...”
- The LED bars should show three indicators. There is an LED on the Right LED unit that indicates that the listing is for DEMO mode. Later, you might have some songs loaded into EEPROM and need to know whether you are in DEMO mode or not. The First LED is the selected song, and the middle LED is the last song in the list (how many demo songs there are).
- Put on your headphones and press “Play”. The LEDs will turn into a progress bar and after it completely loads the song, you should hear some music. You might need to adjust the gain and volume until you can hear the sound. The output mono, it is the same for both left and right.
- After the song ends, use the Left and Right buttons to move the selection to one of the other slots to select another song.
- Press Play to load the song and output the audio to your headphones
- The LED bars should show the progress of the song loading into memory. It will play after loading.
- Adjust the volume so that it is comfortable when listening.
- The Gain knob modifies the preamplifier levels from the synth. The tone of the notes might be distorted if you have it too high, and the VCA might sound less crisp. Adjust it to your liking.
- Try listening to all the demos! These songs were from community arranged sheet music downloaded as MusicXML from Musescore.com, and converted with one of the scripts available on GitHub.

I’ve listened to all of the songs, what else can I do?

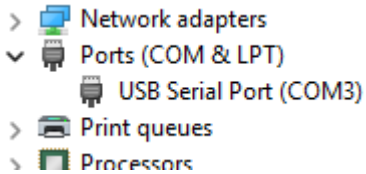
Now comes the part where you feel creative… Open your serial terminal app to the serial port you found before and hit return and couples of times. See the pictures below.

Use the following settings:

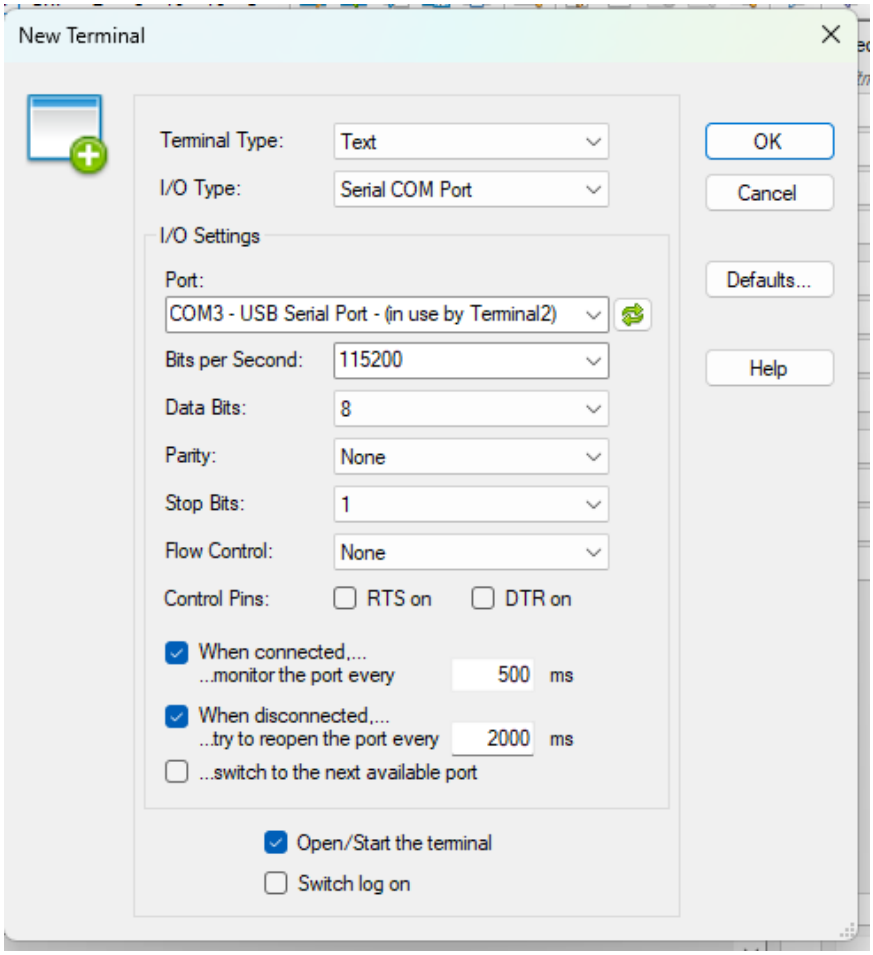
Serial Port:	The port you wrote down
Baudrate:	115200
Flow Control:	None
Data Bits:	8
Parity:	None
Stop Bits:	1
If Applicable	Read/Write so you can send/get messages



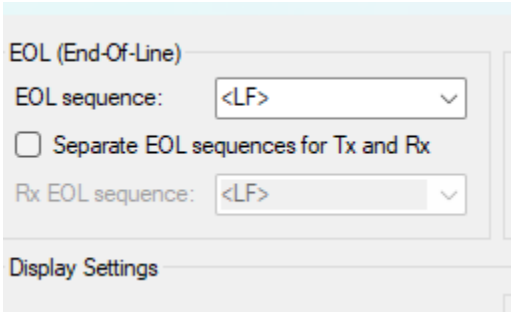
Cutecom on Linux



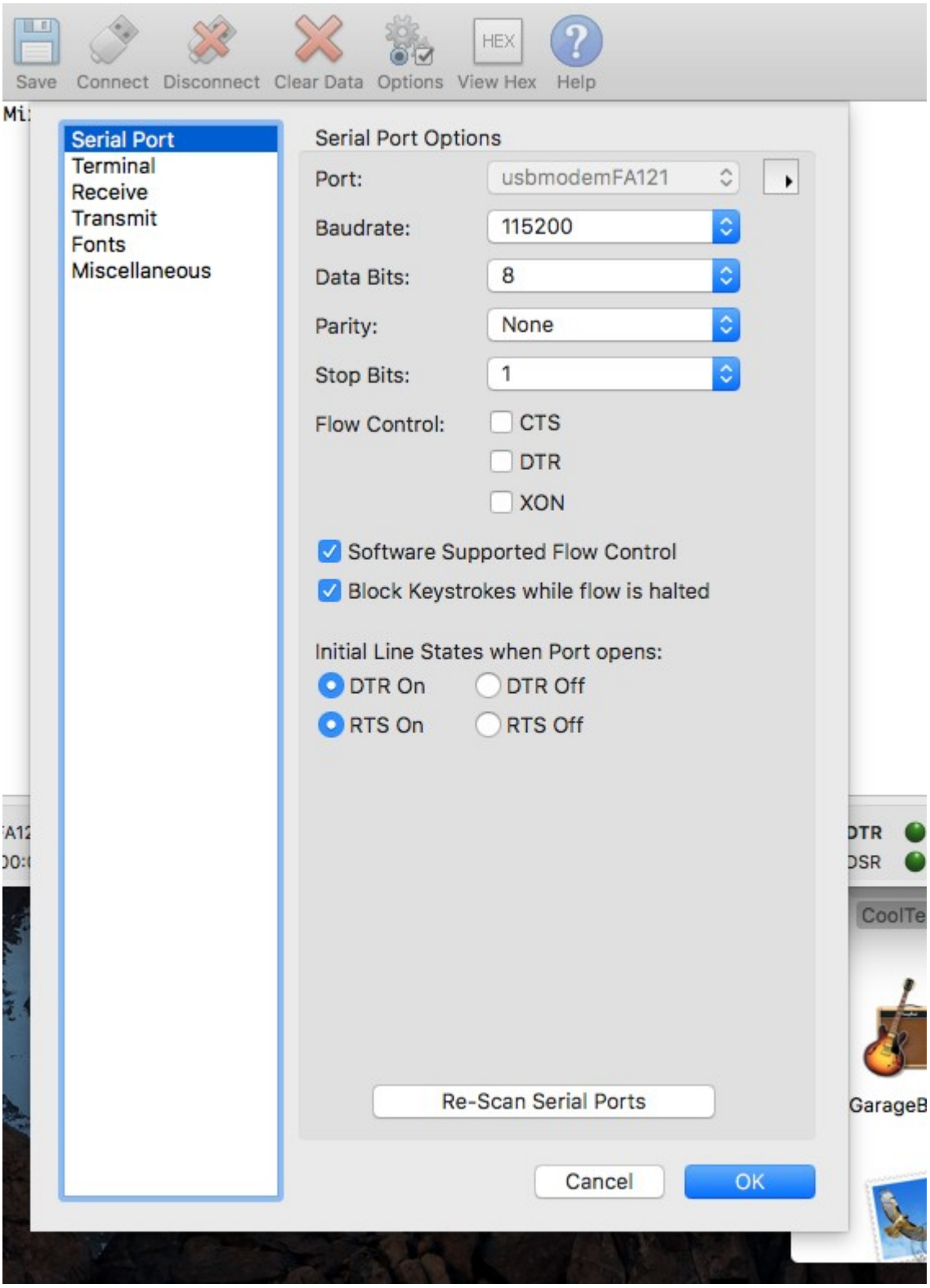
Find USB Serial Port in Windows 11 Device Manager (In this case, COM3)



YAT Connection To Serial Port (COM3) Windows 11



YAT End of Line Settings (LF)



CoolTerm on Mac OS X Sierra

Connect, and hit “Enter” a few times, you should see a command prompt:
“Welcome to Mixtape!” “[mixtape]”

If the output is garbled, you might need to check your BAUD rate, it is hard coded to 115200. Hit enter a few times to get the latest prompt.

For help, enter “?” or “HELP” and press “Enter”. Always press “Enter” after a command to send it through.

You can load the demo songs from the command line and show the code behind them.

To load the fist DEMO song with the command line, enter “DEMO 1”. The LEDs will show the progress until it loads completely.

“Loading DEMO 1… Done!”

The prompt “[mixtape]” should be there, waiting for your next command.

Try typing “PLAY”, and the song should play into your headphones, hmm, interesting, it’s like a tiny computer from 1982… I can type stuff into it to make it do stuff.

Ok… The song is no longer in memory once it is played. Load DEMO 1 again with “DEMO 1”. Now at the next prompt, try typing “BEAT”. It should tell you what the count for 1 beat is (at the time this was written, DEMO 1 had beat of 140). Try entering “BEAT 90”. It should respond “BEAT set to 90”.

Now type “PLAY” at the next prompt… Hey, it is the same song, played faster! Maybe try a couple of values? “BEAT 100”, “PLAY”, and so on. If you want to break in the middle of the song (maybe it is taking too long?) Press the “PLAY” button to exit while playing.

So what if you want to experiment with some other songs?

So the experimental part continues – how do you get new music onto your Mixtape? It all starts with a note! So what is a note comprised of?

Load one of the demos into memory with the “DEMO” command. Type “LIST” to show the song notes. You should get a bunch of “ADD XXXXXXXXX” lines, with the X’s being Hexadecimal numbers. You could output the listing to a log file and edit it if you wanted to change a note here and there from a song in memory.

To Mixtape, a note is composed of the following information:

- Duration
- Voice
- Octave
- Pitch

- Duty
- Initial Volume
- Sustain Volume
- Decay increment/decrement

The duration of a note is how long it is “On” or being played. It is always a whole positive integer number (no fractions or decimals), between 0 and 31. So what if I want an “eighth note” or something more exotic? You will need to use a little bit of math.

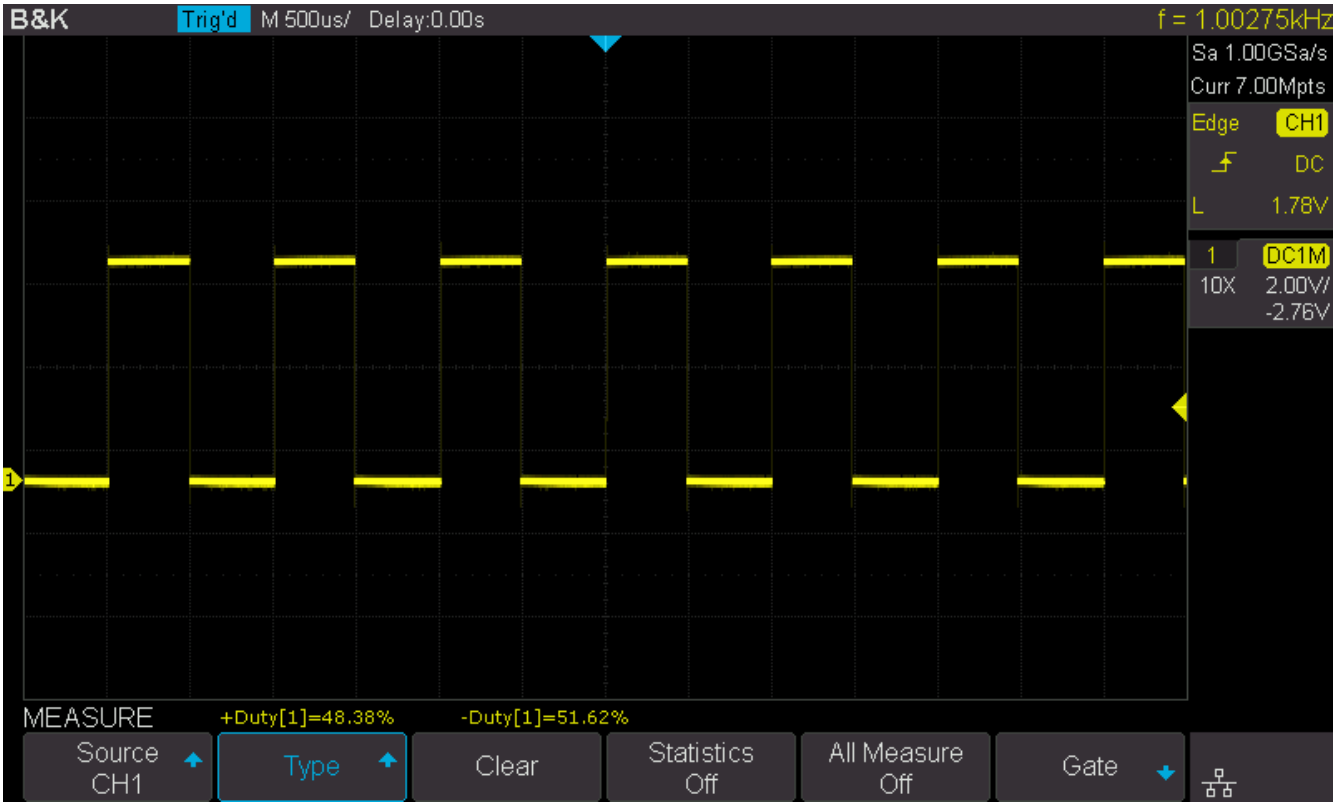
If we were to convert some music to play on Mixtape, we would need to find the shortest note in the entire song (maybe a sixteenth note or shorter?) and that would be duration “1”. All other durations would be a multiple of that note. So, in theory, each measure of music will contain the same number of single durations (I think jazz is when this can really get wild, or look at a Chopin piece). So, in 4/4 time, there are 4 quarter notes per measure, or 1 whole note. If the fastest note is an eighth note, it would receive a value of 1. a quarter note would be 2, a half note would be 4, and a whole note would be duration 8.

The voice of the note should be 1 to 4, but we use the range 0 to 3. You can have 4 part harmonies but need to know which voice to assign the note to. You might want to start with 1 voice then try more.

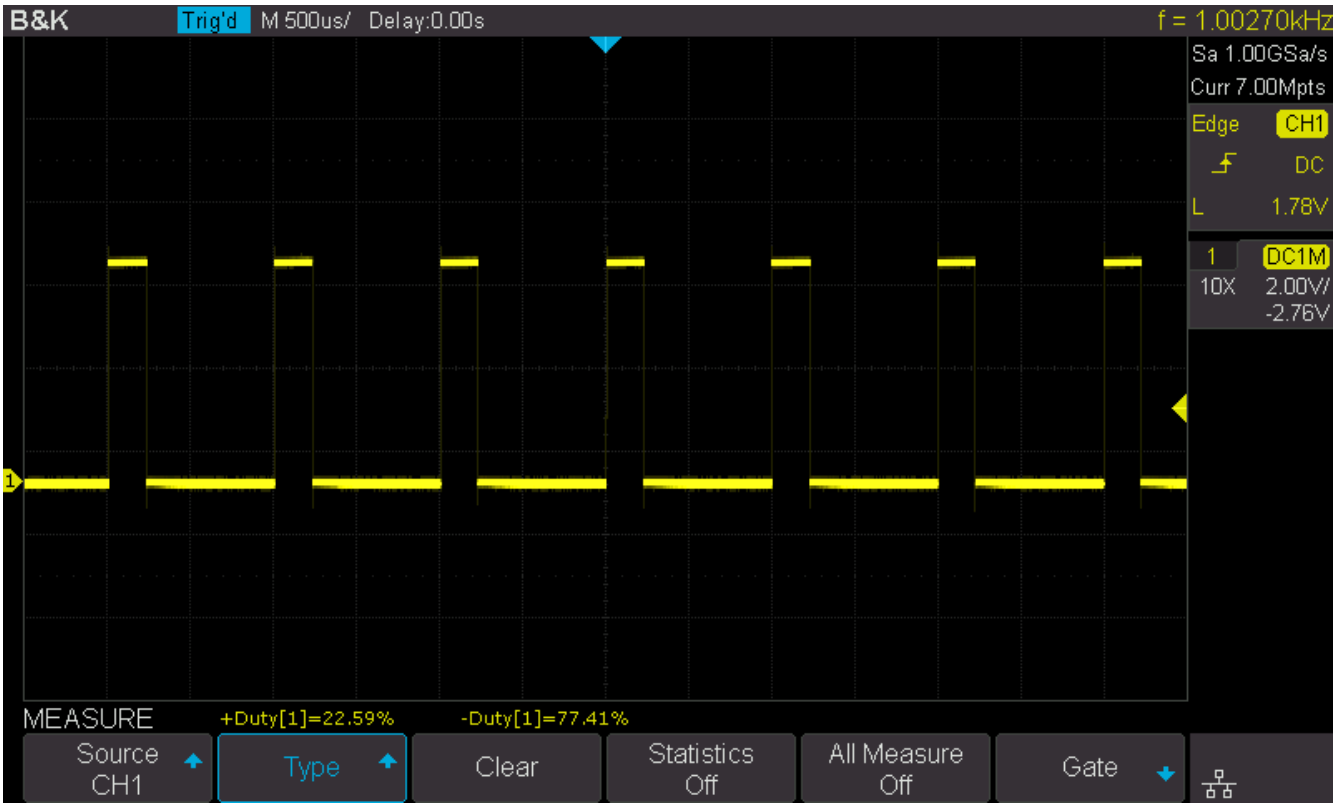
The octave range of this synthesizer is from 1 to 9 (coded as 0 to 8). The higher notes might give you a headache!

The pitch is C to B, in half steps, from 0 to 11. Use pitch = 12 for a rest note (no sound, zero volume).

To understand the duty, you might need to understand a little bit of physics/math about a waveform. A note is a periodic wave; it could be a sine/cosine wave, a triangle wave, ramp wave, or in this case, a square wave. Take a look at the oscilloscope image below to see a wave with 50% duty. You can see a full period goes from 0 Volts to 5 Volts to 0 Volts again. The part of the wave 0 to 5 volts is the same length as 5 to 0 volts, so it is 50/50. The sound is the most hollow sounding when it is 50/50. A more tinny sound happens when you shorten one side of the it. The duty value can be 0 to 15. The value 7 is 50% ((7+1)/16) = 0.50. So a value of 3 would be ((3+1)/16) = 0.25, or 25%/75%. 0 is reserved for a drum (future feature).



Approximately 50% duty cycle



Approximately 25%/75% duty cycle

Next comes information about the ADSR. What is ADSR? It stands for Attack, Decay, Sustain, Release. You might want to read up on it on Wikipedia for a summary. For Mixtape, you have the duration of a note as mentioned before, and a volume level of 0 to 15, with 15 being loudest. Now divide the duration by 4 in your imagination – the first 2 out of 4 (half) of the duration will be the initial volume. Section 3 out of 4 will be the sustain volume. Section 4 out of 4 is divided into 8 sections, and each section will decay from the sustain volume by the value “decay”.

Gimme a “C”, a bouncy “C”…

Let’s start with a scale!

1 Beat per note

Only Voice 1

Octave 3
C, D, E, F, G, A, B,
Octave 4
C

50/50 Duty

15 Initial volume
11 Sustain
2 Decay

We have code for this! To shorten the amount of typing you’ll need to use Hexadecimal notation for 0 to 15 (0 to F). Hex, or Base16, uses 0-9 for numbers 0 to 9, but for 10, it uses “A”, 11 is “B”, 12 is “C”, 13 is “D”, 14 is “E”, and 15 is “F”. Convenient because 4 bits can be represented as one digit, 8 bit = 2x4 bit.

The code for a note is:

{duration}{voice}{octave}{pitch}{duty}{initialvol}{sustainvol}{decayval}

The first “C” in our scale is:

{01}{0}{3}{0}{7}{F}{B}{2}

The ranges for each code are:
Duration is two hex digits, but 5 bits, giving you 0-31.
Voice is 0-3
Octave is 0-12 (0-C)
Pitch is 0-13 (13 is rest note) (0-D)
Duty is 0-15 (0 is for Drum, not yet implemented) (0-F)

InitialVol is 0-15 (0-F)
SustainVol is 0-15 (0-F0
DecayVal is 0-3

The brackets are just to visualize it, so remove the brackets and you can copy paste the whole thing 8 times and change the 1 or 2 values to get (don’t type this in yet)

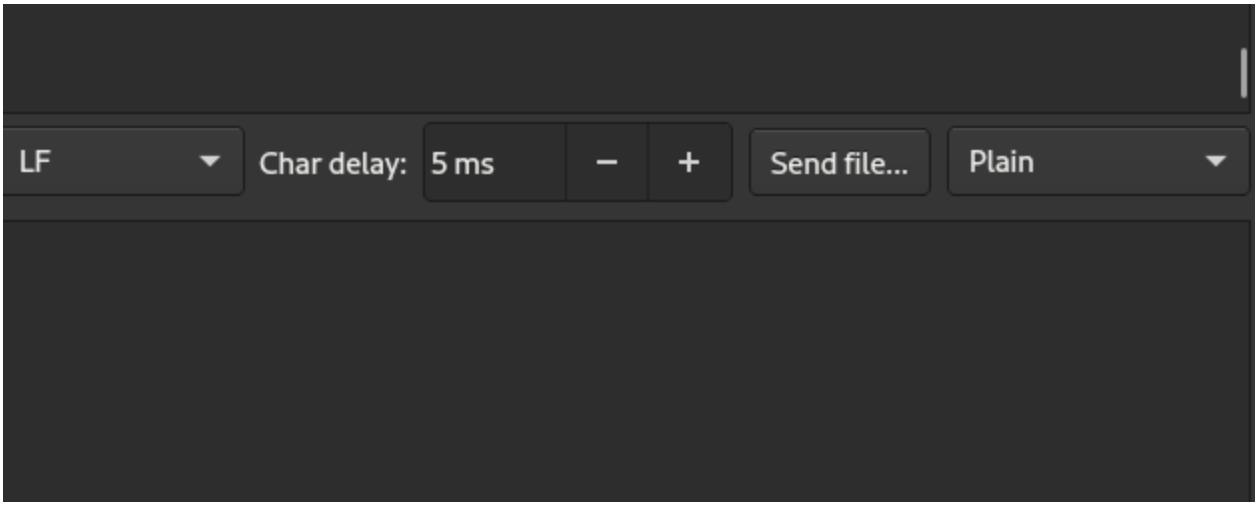
010307FB2

So it follows that the other notes (skipping sharps/flats):
010327FB2 (2 in 4th digit is D)
010347FB2 (4 in 4th digit is E)
010357FB2
010377FB2
010397FB2
0103B7FB2 (B in 4th digit is 11, or B)
010407FB2 (Increased an octave, 4 in 3rd digit, 0 for C in 4th digit)

Let’s enter it on the command line:

```
CLRR
ADD 010327FB2
ADD 010347FB2
ADD 010357FB2
ADD 010377FB2
ADD 010397FB2
ADD 0103B7FB2
ADD 010407FB2
BEAT 120
PLAY
```

You can add it to a text file in your favorite code editor, or in a notebook and can send it to Mixtape as a file over your serial monitor after making changes. You might need to add a delay in CuteCom to make sure that the text isn’t sent over too quickly for the device to recognize it. 5 milliseconds seems to work, at least for my computer.



You can save the current memory and beats to EEPROM, which sticks around after you unplug the device. Enter your scale as before, or send it over by file.

Then you can save it by typing “SAVE”. It should save to the next memory location. You can toggle demo mode with the same command as before (hold middle button hold left button release middle, left). You should see one LED lit for your newly saved scale. Press play to load and play it. You can also update the beats that were saved by entering a new BEAT value, and then UPDATE 1 (or whatever song index it is).

To list the saved songs, type in “HELP” and it will show you all the songs and their beats.

To load the song on the command line, type “LOAD 1” replacing the 1 with the correct index. You can then type in “PLAY” to play the song in memory.

Now that you know how to enter notes and move around, if you are a maker of sorts, a programmer, or have some coding knowledge, you can do all sorts of things. The code is quantifiable, so maybe you can teach AI how to make notes that are sequenced and arranged to sound nice? Just a thought! Before we go there, you can download songs off the internet in a number of formats and convert them into Mixtape format. If the AI thing happens, you can make it spit out some code and you can upload it to see if it is the next big hit. Some scripts are on Github that you can modify for fun...

Make it easier, please…

OK, there are some scripts that can make the music conversion easier. Go to GitHub and clone the following repository:

https://github.com/ramonricca/mixtape

In the scripts directory, there is a script to convert MusicXML into Mixtape format. You will need to install a few perl modules for it to work.

On Ubuntu, install:
sudo apt install libxml-xslt-perl libxml-simple-perl

Windows doesn't have perl installed by default, please download "Strawberry Perl." You might need to google how to install modules or check stack overflow.

In the "sampleXML" directory, there are some songs to try to process.

cd scripts
chmod u+x readxml_measure.pl (once to make it executable)
./readxml_measure.pl sampleXML/odetojoy.xml

you should see a bunch of "ADD" note statements.

./readxml_measure.pl sample/XML/odetojoy.xml > ode.txt

Now you can send the ode.txt file over to Mixtape using CuteCom.

Where to go from here…

Some future iterations of Mixtape might include other effects like drums. But if you want to experiment with your own sound filters or process the sound with more effects, you can use a patch cable from the headset jack to a computer and use some free software like Audacity to mix it with other ideas. Also, if you have a soldering iron and some ideas for circuits, the circuit board has a header with a pin for GND, mixed headphone out, all 4 voices with and without VCA. Solder a header onto the board, and then you can use jumpers with a breadboard to design a circuit. You can use it as the basis for your own modular synthesizer! If you are looking for a project you could write a plugin for lilypad or musescript studio that converts sheet music directly into Mixtape format.

You can make Mixtape better by reporting bugs on the GitHub page or by emailing me directly. Mixtape's firmware can be update by USB using a python script -- your suggestions can be pushed out to all the users that receive this Proof of Concept. You can also fork the repo on GitHub and share updates to the scripts.

Additional notes (last minute):

Note that the commands are not case sensitive and convert to upper case before execution. Also, all characters after a pound symbol "#" are considered comments, and are ignored. So you can use it to comment up a song if you wish. Also, a command line is maximum 40 characters, or else it restarts the array. Using the "LIST" command with a number after it tries to convert the notes in memory back to

measures, so “LIST 16” will printout the voices in measures of 16 beats. You can update the beat set in EEPROM to the current beat by using “UPDATE {index}” after setting the beat with “BEAT {n}”. CLRR will clear the song in RAM. To clear the EEPROM, use “RESET”. Type “HELP” for a summary of available commands.

IMPORTANT! If you download MusicXML files from the web, they are often compressed, and have the extension .mxl. To use these scripts, unzip the .mxl file by using “unzip filename.mxl” on linux, or use Windows/Mac built in archiving unzip. Then the scripts might have a chance of working on the sheet music.