



# GRAINS

*uploading an Arduino code*



# Introduction

Your Grains module works with an Arduino Nano board and needs to have a code uploaded. We will show you how to do it.

For this instruction we used the Arduino IDE 1.8.13 running on an Apple macOS Catalina machine. But a lot of the steps will match with those needed by different versions of operating systems and/or Arduino IDE.

Important: Make sure you have a fitting data cable to connect your Grains to your computer. A charging cable won't work. You need a USB A male to USB mini B male cable.

## Preparing

### **Download and unpack the Grains codes**

Turn on your computer, make a folder with the name Grains or another recognizable name, open your browser and go to <https://www.ginkosynthese.com/grains-codes>, download your zip of choice and put it in your new folder. Unpack the zip.

### **Download and install the Arduino IDE**

In your browser, go to <https://www.arduino.cc/>, choose Software -> Downloads.

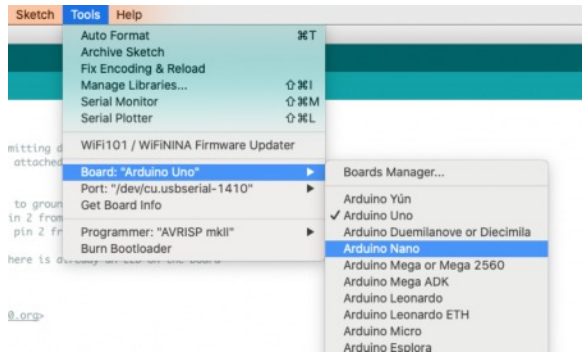
Scroll to Download the Arduino IDE section, download the file for your type of operating system and put it in your new folder. Unpack the zip (Mac) or install the program (Windows), and close it.

*Windows users, if you have no Arduino experience it is the easiest way is to use the Windows installer (as admin), so ignore the non admin and app installers.*

### **Download and install the CH340 driver**

The Arduino Nano board used in the Grains is a Chinese one. It needs another driver than the one already installed with the Arduino IDE. Go to <https://sparks.gogo.co.nz/ch340.html> to download the driver for your system, and install it.

# Attaching the Grains Nano board to your computer



Start up your computer, attach the Grains'Arduino Nano board to your computer using a data usb cable. Open the Arduino IDE program. You will download the code later, first you have to select the right board, processor and COM port.

Go to Tools-> Board -> Arduino AVR board and click Arduino Nano.



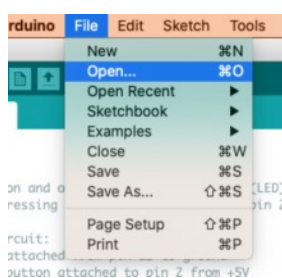
Stay in Tools, go to Processor and choose ATmega 328P (Old Bootloader). Most problems with a failing upload are due to choosing the processor type with the wrong bootloader.

Again in Tools, go to Port and click on the COM port you think the Nano board is attached to.

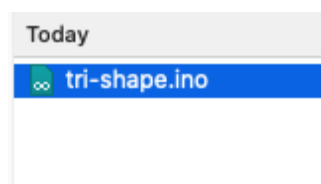
Now click on Get Board Info in the Tools menu. If you made the right choice of board and COM port, an information window will pop up. Note that successfully uploading a sketch is the only way to check if the right processor was selected.

Mac users: also note that the COM port probably can't be tested if it's a native port.

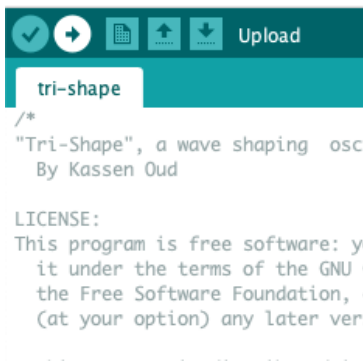
## Opening a the code file in the Arduino IDE



Next step is opening the code in the Arduino IDE program. Choose File -> Open and go to the folder where you unpacked the code you chose. Open the folder containing the code, and click on the .ino file. This will open a new IDE window showing the code.

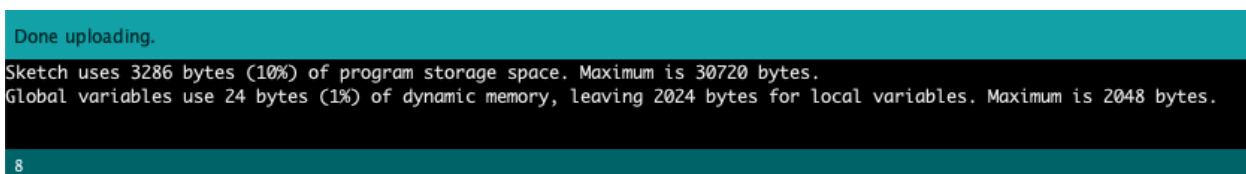


# Uploading the code



You can upload the code to the Arduino Nano board by clicking on the arrow pointing right, in the dark green bar on the upper left of the IDE.

In the black window at the bottom you can see the protocols that the program is executing. In the teal bar above this window you can follow the progress. When the uploading stops and the status is Done Uploading, you are ready to detach the Grains from the computer and put it in your modular.



# Troubleshooting

## Board connection

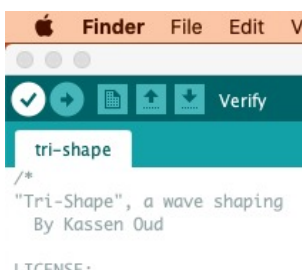
When the Arduino Nano board is not recognised by the Arduino IDE: make sure you use a data cable and not a charging cable, check if your cable is working at all. In the IDE: make sure you have chosen the right board, processor (mind the different bootloaders), and COM port. Try different COM ports.

## Upload doesn't work

Make sure the board is connected correctly. Again, mind the type of processor combined with the bootloader. Change it to see if the other combination gives a better result. Make sure you tried to open the .ino file.

## In general

You can verify if the code is compliant by clicking on the first button in the dark green bar, upper left of the IDE.



## Error messages

Error messages in the IDE show as orange text in the black status window, and can be copied for search engine use by clicking the Copy Error Message button.

```
Problem uploading to board. See http://www.arduino.cc/en/Guide/Troubleshooting#upload for suggestions.
avrdude: stk500_recv(): programmer is not responding
avrdude: stk500_getsync() attempt 10 of 10: not in sync: resp=0x00
Problem uploading to board. See http://www.arduino.cc/en/Guide/Troubleshooting#upload for suggestions.
14
```



## Extra: look for info in the code

In the Arduino .ino files often a comment section is included. This section is very interesting as it can contain a short description of the script, and a manual on how to use it. The comment section is the text between the /\* and \*/ marks.

```
tri-shape
/*
"Tri-Shape", a wave shaping oscillator for Ginkosynthese's "grains" module.
By Kassen Oud

LICENSE:
This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program. If not, see <http://www.gnu.org/licenses/>.

DESCRIPTION;
A triangle-based oscillator with wave-folding and hard-clipping features.

MANUAL;
Knob 1 / mod 1; Wave folding
Amplifies the base triangle wave, then once it hits the limits of the
range "folds" it back in the opposite direction. With enough
amplification this will happen again at the other side, creating a
rich spectrum.
Knob 2 / mod 2; Hard Clipping
Amplifies the signal (after the folding) and clips the result. This is
done asymmetrically for a "deeper" and more "bassy" sound. Thanks to
Rob Bothof for that trick.
Knob 3: Base Tuning.
Sets the base pitch.
Mod 3; Pitch Modulation
This CV gets added to the value set by Knob 3.

*/
```