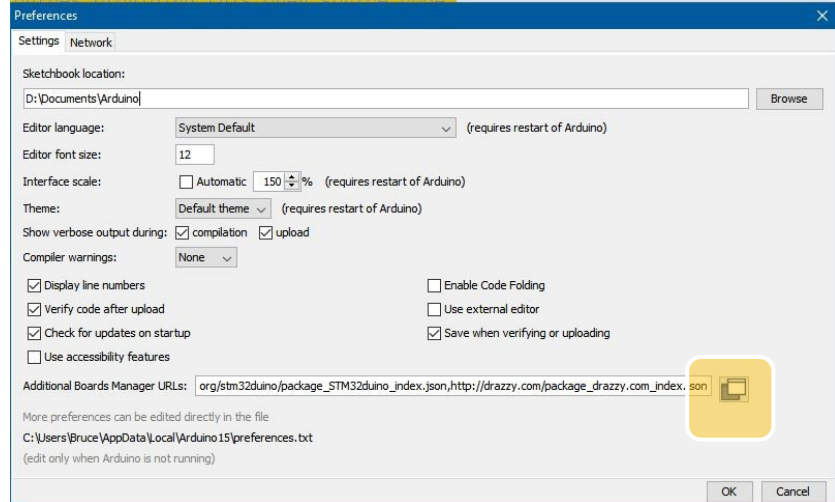


Setting up the Arduino IDE.

Start your Arduino IDE. Go to Help->Arduino to check the software version you are using. The number will be inconspicuously written at the top of the opened window. My current version is 1.8.13.

Go to File->Preferences. Your screen should look something like this:



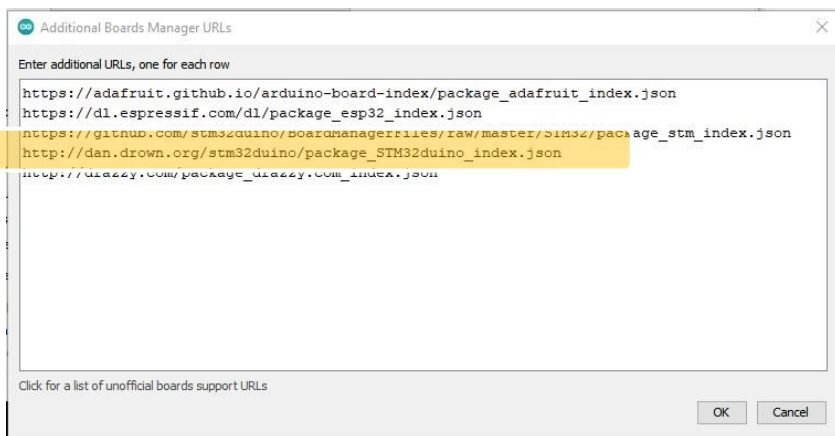
Your sketchbook location is where Arduino will store your sketches (and Libraries). I store mine in a subfolder of My Documents.

Now look at the box labelled “Additional Boards Manager URLs”. The Arduino IDE needs to be told specific information for each hardware device that it programs, whether it be an Arduino Nano, a “Blue Pill”, an ESP32 board, or something else. The entries on this line tell the Arduino IDE where to look for this board information.

Click the “Double-Window” icon, highlighted above, to see all of the URLs. My list looks like this:

I program a lot of Adafruit boards, so the first line adds all of the Adafruit boards.

The fourth line, “dan.drown.org”, loads drivers for the Blue Pill and related STM32 boards. These particular drivers were written by good programmer named Roger Clark.



Make sure your list includes the dan.drown.org line. This is the only line needed.

By the way, you can also add board files manually by adding them to a subdirectory of your sketches folder. This is only necessary if you do not have a corresponding “package” for the files, like the one above. Since we have a package for Roger Clark’s STM32 core, we do not need to add them manually.

Libraries Overview

Libraries provide low-level hardware interfaces. Specifically, Morse Tutor uses two libraries from Adafruit for writing to the display, and another library for reading from the SD card. These libraries are stored in a subdirectory of your sketches folder. On my computer, my sketches folder is D:\documents\Arduino. Any sketches I create are stored in this folder, and any libraries I add to my system are stored at D:\documents\Arduino\libraries. Over time, this folder may grow to contain tens to hundreds of libraries. To remove a library, simply delete its folder. In fact, you can remove all of your old libraries by deleting the entire libraries folder. A nice explanation of libraries is found [here](#).

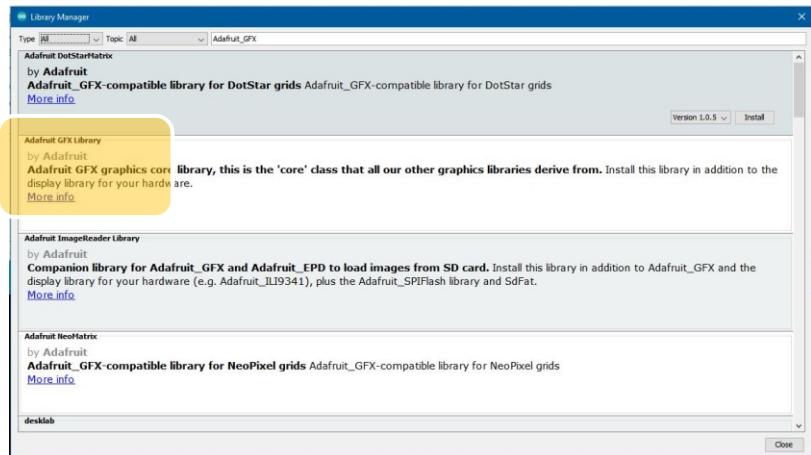
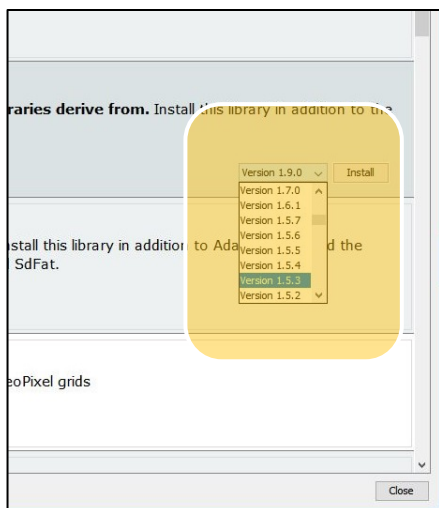
The code contained in each library may change over time, as the creators and contributors of the library refine the code. For this reason, it is important to know the version numbers of the libraries that you are using.

To manage your libraries from the Arduino IDE, and see which versions you have installed, go to Sketch->Include Library->Manage Libraries or press Ctrl-Shift-I. To see the libraries you have installed, Change the drop-down box labelled Type to “Installed”. A brand-new Arduino installation will include libraries without any action on your part. In version 1.8.13 there are 19 built-in libraries for your use.

The Library Manager includes references to the most popular libraries, making it easy to add them to your system. For example, change the type to “All” and type Adafruit_GFX in the box labelled “filter your search”:

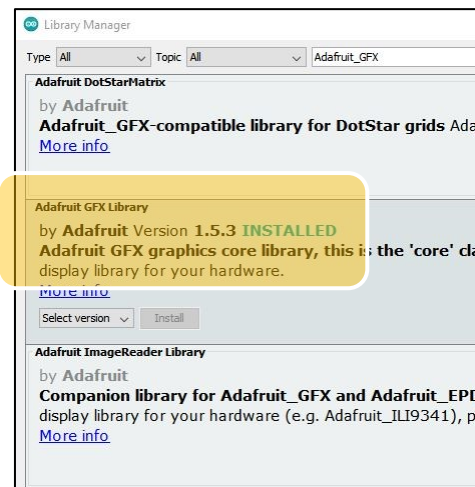
You will see several choices as shown. The second choice

“Adafruit GFX Library” sounds like the right choice for our project, and it is. If you hover over this choice with your mouse, you will notice that a dropdown box will appear in the lower right of the panel, listing



the available versions to install. For our project, we want version 1.5.3. Select this version then click the Install button.

Once a library is installed, the word ‘INSTALLED’ and the version number will appear in its listing



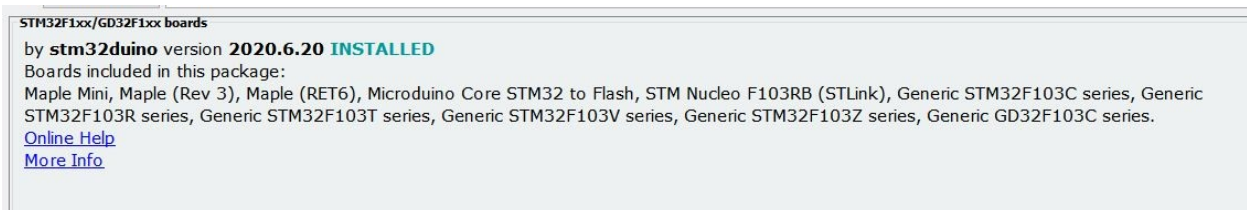
A confusing aspect of STM32/Blue Pill programming is that there is more than one available “core”. A core is a software package that provides an interface for your specific board. There are two widely-used cores:

1. Roger Clark’s core: this is the one described above. It offers optimized STM32 code, especially for the Blue Pill/Maple clones. The big disadvantage is that Roger no longer actively maintains the code.
2. The official STM32 core from STmicroelectronics: It offers broad support across many different boards and STM32 microcontrollers. The disadvantage is poor performance, and compatibility with other libraries suffers.

Over time, I suspect that most users will move from Roger Clark’s core to the official one. Unfortunately, the official core is still incompatible with several libraries that I use.

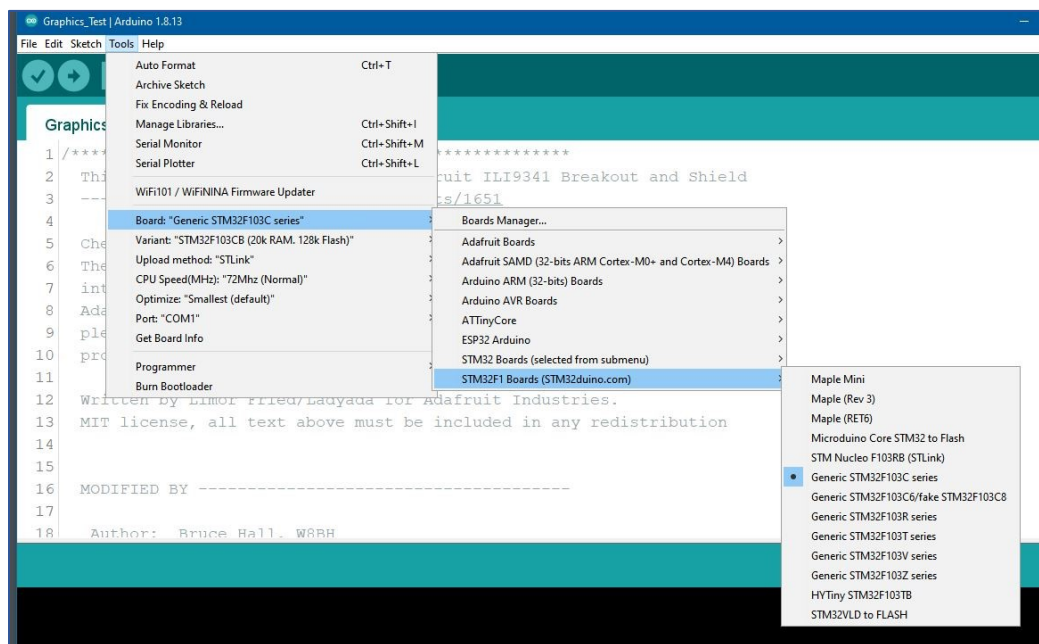
Board Setup

In the Arduino IDE, go to Tools/Board/Board Manager and search for “STM32”. If you installed the dan.drown.org package correctly, you should see an entry for STM32F1xx boards by “stm32duino”. This is the Roger Clark core:

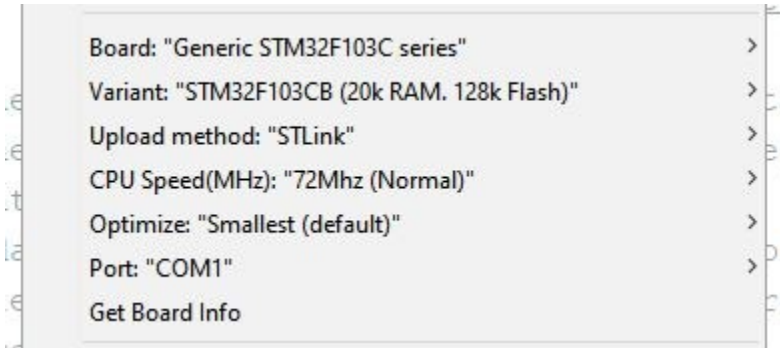


Now go back to the Tools Menu and Select “Board”. From the available choices, select STM32F1 boards, and then “Generic STM32F103C series”:

For Variant, you can select either 64K or 128K,



depending on your chip.



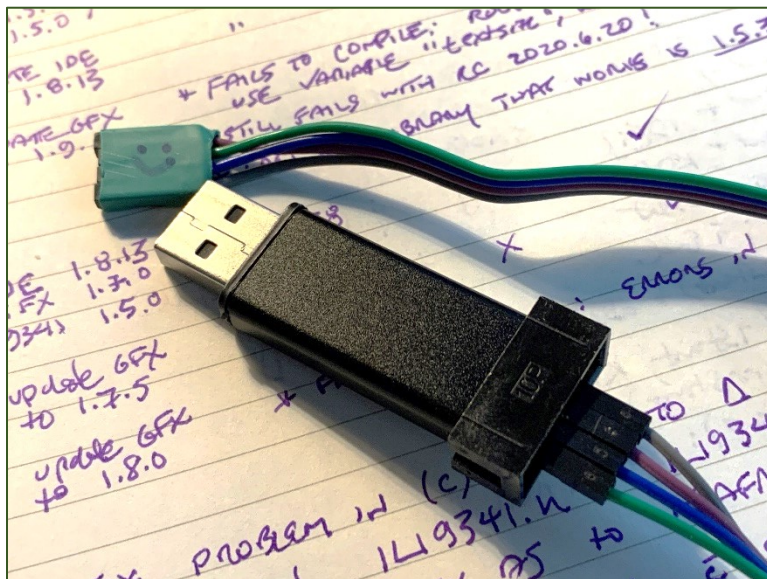
Most Blue Pills are specified for 64K but actually have 128K.

Finally, choose ST-LINK for the upload method. Your board settings should look like the illustration at left.

ST-LINK

ST-LINK is the fastest and easiest way to program Blue Pills, in my opinion. The official ST-LINK/V2 hardware can be obtained from Mouser and Digikey for about \$20. Clones can be had on eBay for about \$3, and come with a small female-to-female cable. One end goes into the ST-link pins 2,4,6,8 (SWDIO, GND, SWCLK, 3.3V), and the other end of the cable goes onto corresponding pins on the Blue Pill.

Reconnecting individual pins is error-prone, so I bound the pins together with a dab of blue and heat-shrink tubing. I also marked the face of the plug so that I know which side is up.



On final bit of advice: make sure that you board is powered before attaching it to ST-LINK. An unpowered board will draw current from the ST-LINK device and could damage its small on-board voltage regulator.

Good luck with your Blue Pill programming!

73, Bruce

W8BH.net