LEARN

Arduino Basics: Library & Board Setup

With 8BitCADE

Compatible with:

8BitCADE XL

8BitCADE.com

Contents

Written by the 8BitCADE Team

Support@8bitcade.com

Version 1

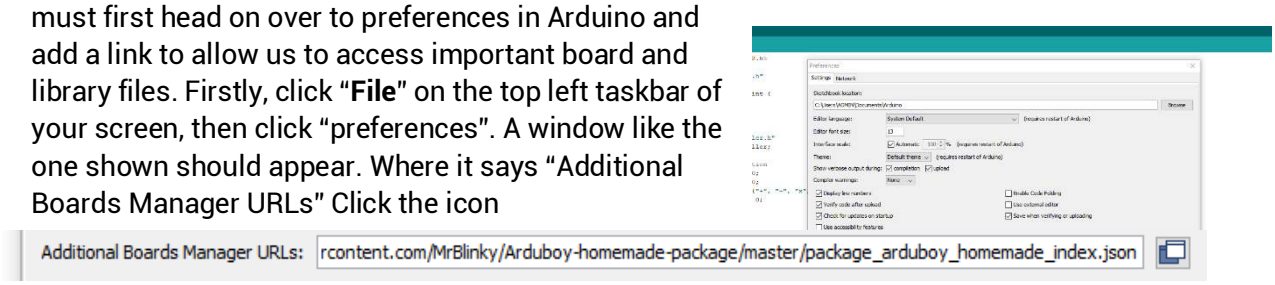
Setting up

In this tutorial, we will go through the setup required to get the libraries and board settings we use for the 8BitCADE XL ready. If you want to upload premade games, follow the "8BitCADE XL Uploading Games" Tutorial.

Board/Library Install

Before we write code, we need to include a specific library that will make programming your 8BitCADE much easier. The Arduboy2 Library. To begin this setup, we must first head on over to preferences in Arduino and add a link to allow us to access important board and library files. Firstly, click "**File**" on the top left taskbar of your screen, then click "preferences". A window like the one shown should appear. Where it says "Additional Boards Manager URLs" Click the icon

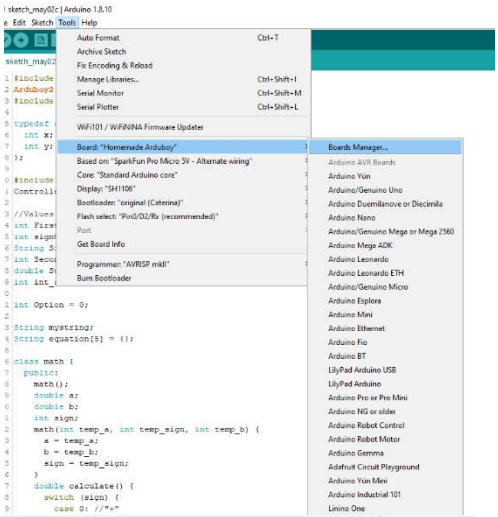
```
1 #include <Arduboy2.h>
2 Arduboy2 aboy;
```



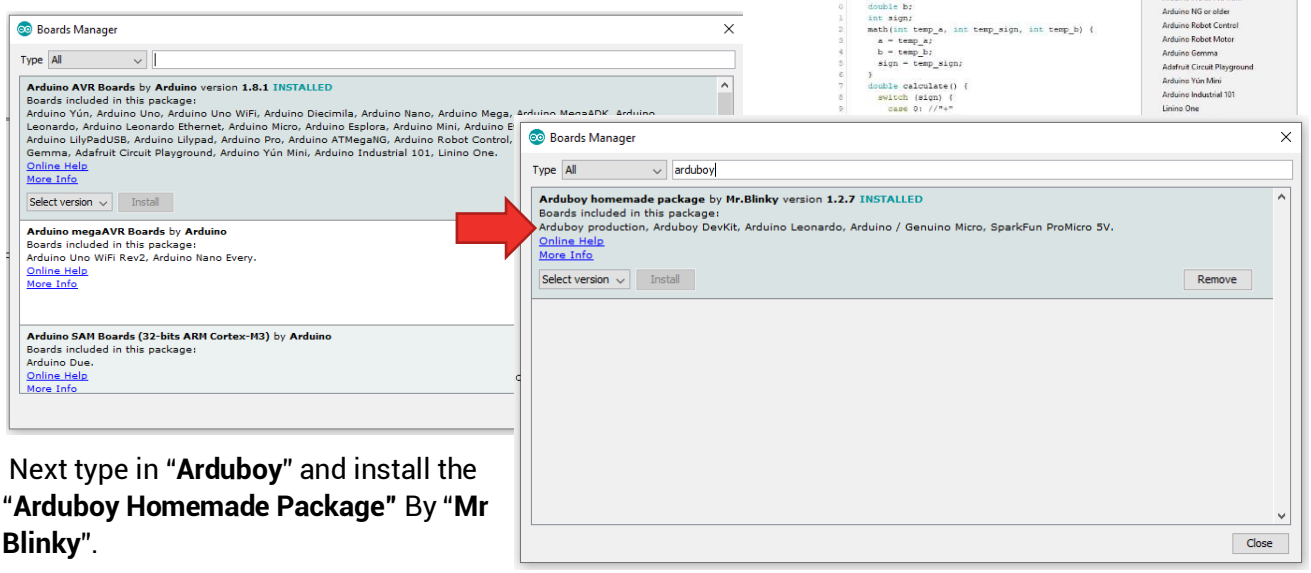
And type in, on a new line:

https://raw.githubusercontent.com/MrBlinky/Arduboy-homemade-package/master/package_arduboy_homemade_index.json

This will allow us to access the board and library information. Next, we need to install the board and all of its libraries. To do this, exit the current window and click "**Tools**" and select "**Board: […]" > "Boards Manager**"

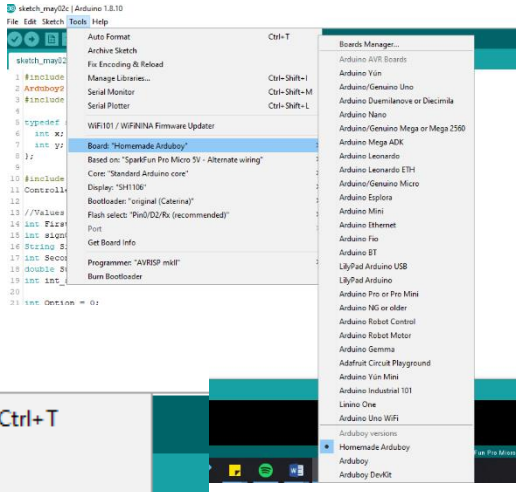


The below window should pop up (it will take some time as all your libraries are being checked/updated if need be).

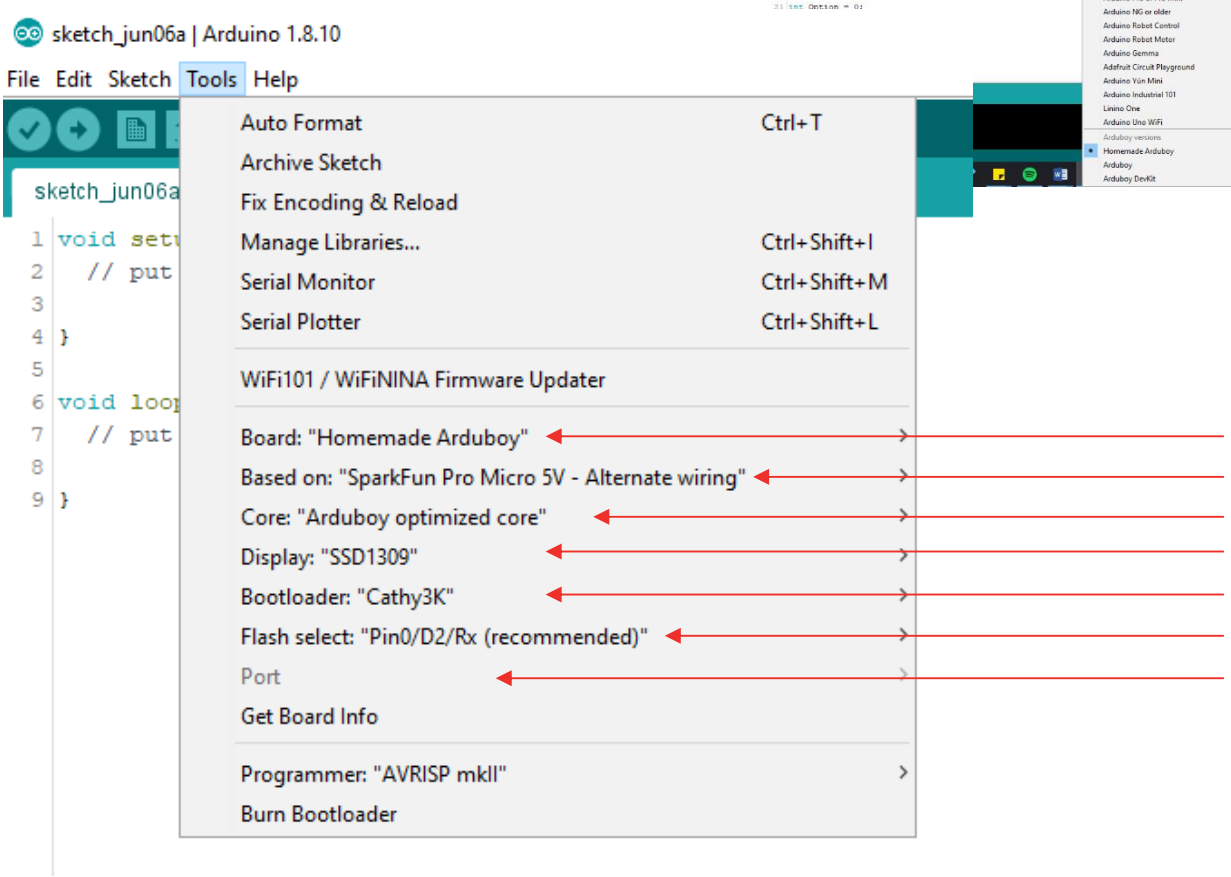


Next type in "**Arduboy**" and install the "**Arduboy Homemade Package**" By "**Mr Blinky**".

Next, we need to head on over to select the board we just downloaded. To do this head over to the toolbar and click "**Tools**" and select "**Board: […]**" > "**Home Made Arduboy**"



The next step is important, and you should double-check your settings. Check and change your board values to be exactly like the photo below:



The port will be set to whatever "COM" your console is connected too.

Using the Library

Whenever you are using a library in Arduino, it's important to include it in your sketch. To include a library simply write:

```
#include <Arduboy2.h>
Arduboy2 aboy;
```

#include <LibraryName.h>

In this case, we are including the ArduBoy2 Library, one of 6 library's that we can use. After including the Arduboy library, we can redefine its name. Instead of writing "Arduboy2.[Function]()" etc, we can type "aboy. [Function] ();"

Test Code

```
#include <Arduboy2.h>
Arduboy2 arduboy;

void setup() {
  // put your setup code here, to run once:
  arduboy.begin();
  arduboy.clear();
  arduboy.print("I Love DT!");
  arduboy.display();
}

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

}
```

The above code should run without any errors and produce the above output. Be sure to run this to test that both your screen is working and that the library is correctly installed and that you can use Arduboy Functions correctly

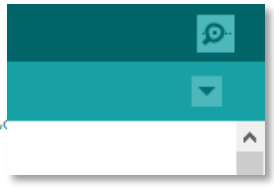
Using Serial

Before we go any further – it's important to understand how to use a command called 'Serial', and many functions such as begin and print.

Serial is a way for our computer to communicate with our Arduino – all Arduino boards have at least one 'Serial Port' which consists of both an RX pin (used to receive data) and a TX pin (used to transmit data). To communicate with our Arduino, we must first use the **Serial.begin(Speed)** command – where we define the rate of bits per second at which we will transmit data, this is known as a baud rate – for those interested in more, check out this link here.

In our case, we will be using **9600**. Once the Serial has been initialized viz the being command, we can use various commands to transmit or receive data from our Arduino – in this case, we will use **Serial.print()** (or **Serial.println()** if you want to print on a new line) to display the data in our classes. Another use for serial is for debugging. Print variables to serial so you can analyse the variables and ensure they are the correct expected values.

```
1  void setup() {
2    // put your setup code here, to run once:
3    arduboy.begin();
4    arduboy.clear();
5    arduboy.print("I Love DT!");
6    arduboy.display();
7
8    //Begin Serial
9    Serial.begin(9600);
10 }
11
12 void loop() {
13   Serial.println("I L
14
15 }
```

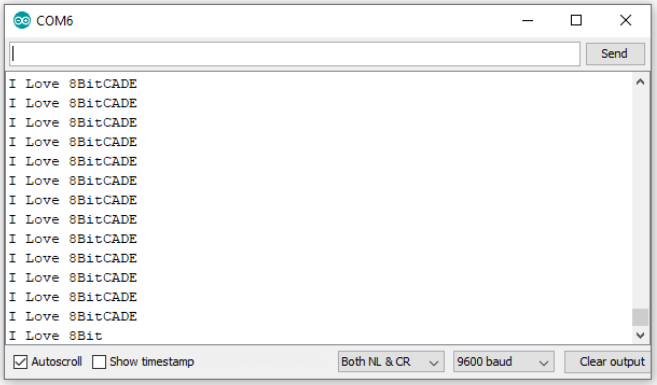


Write the code shown on the left. Here you can see how we use serial to print out a similar message to what we printed on the actual screen.

Now we printed to serial, but how do we even see this data? Where did we print it too? Well, we use something called "Serial Monitor" that can be accessed by clicking the icon shown on the right (this can only be opened if an Arduino is plugged in).

Open the serial monitor and check your monitor to mine on the next page!

Your monitor should look like this:



In the Serial Monitor, we have options such as:

Toggle **AutoScroll**: Use this if the data being printed is being printed too fast and you need to stop and view a certain bit of data (via scrolling up and down)

Toggle **Timestamp**: Will allow you to see when something was printed to serial.

The clear output allows you to clear the current serial monitor.

Finally, we can adjust the Serial Monitor speed/baud rate– this MUST be the same value as the one specified in the Serial.begin().

Going Further With Arduboy

If you want to read more about Mr Blinkys Arduboy Library, then check out this link:

https://github.com/MrBlinky/Arduboy-homemade-package

If you want to check out the Arduboy Library documentation, check out this link:

https://mlxxxp.github.io/documents/Arduino/libraries/Arduboy2/Doxygen/html/classArduboy2.html

This contains all of the different functions we can use.