

CB PLL programming instructions ESP32-S3

Version 17/9/2024 © Jef Collin

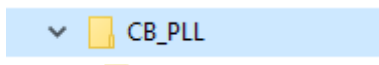
The ESP32-S3 needs to be programmed before use, the INO file holds the source code. These instructions will guide you through the process:

- Installing the Arduino IDE environment
- Copying the source code
- Installing the additional libraries
- Programming the ESP32-S3

The Arduino IDE is a moving target, constant development is done on the IDE and libraries, often there are breaking changes in the libraries and the code no longer compiles.

To avoid these pitfalls I recommend using Arduino IDE version 1.8.19. In portable mode, this allows you to keep all you need in one folder and “freeze” the entire setup in a known working state, this is not (yet) possible with version 2. You could even run the entire setup from an USB stick.

Make a new folder.



Go to the Arduino download page.

<https://www.arduino.cc/en/software>

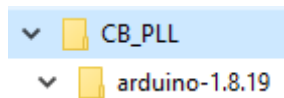
Download the ZIP file for version 1.8.19 (final version of IDE 1), DO NOT download the installer.

Legacy IDE (1.8.X)

A screenshot of the Arduino IDE 1.8.19 download page. The page has a light blue header with the Arduino logo and the text 'Arduino IDE 1.8.19'. Below the header, there is a description of the IDE and a link to the documentation. On the right side, there is a 'DOWNLOAD OPTIONS' section with a red circle around the 'Windows ZIP file' option. Other options include 'Windows app', 'Linux', and 'Mac OS X'. The 'Linux' section lists 32 bits, 64 bits, ARM 32 bits, and ARM 64 bits. The 'Mac OS X' section lists 10.10 or newer. There are also links for 'Release Notes' and 'Checksums (sha512)'.

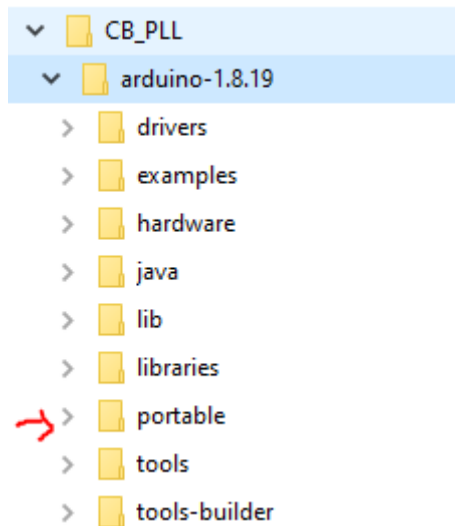
Copy the ZIP file to the new folder and unpack, do not run the arduino.exe software yet (you can delete the ZIP file when done unpacking).

This will make a new folder with the Arduino IDE.



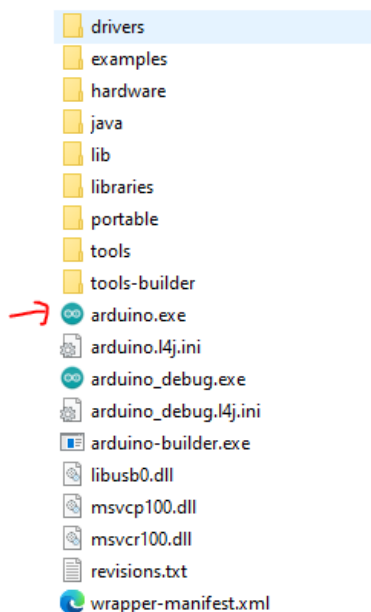
Open this folder, depending on how you unpacked there might be extra folder levels, open until you get the tree structure below.

In this structure create a new folder called “portable”.



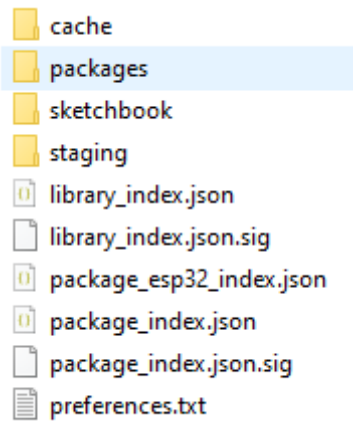
Start the Arduino IDE by clicking *arduino.exe*.

Important: this will be the method of starting the Arduino IDE for this development, do not try to open it by clicking on the INO source file since that will try to open any Arduino IDE you have installed with the installer, since this one is not “installed” Windows does not know the correct one to open. You can make multiple folders with other development setups for other projects.



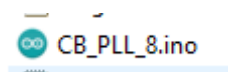
The IDE will detect the portable folder and will create additional folders within it. Wait a moment until the IDE is fully loaded and close it.

Open the portable folder, there will be folders and files created by the IDE.



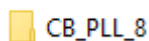
Open the sketchbook folder.

Look at the .INO file in the src folder on github.



Create a new folder with the exact same name as the .INO file in the SCR folder on github, note that the number (in this case 8) can change, I use an new number for every major change of the source code regardless of what the final version number will be.


Create a matching folder, the folder name must match the filename.



Copy the .INO file and all .H (header) files in the github folder to this folder.

arduino-1.8.19 > portable > sketchbook > CB_PLL_8	
Naam	Gewijzigd op
7seg_31_19.h	31-7-2024 15:1
7sega.h	31-7-2024 15:1
CB_PLL_8.ino	2-8-2024 21:14
DIN_1451_Mittelschrift_Regular8pt7b.h	31-7-2024 15:1
DIN_1451_Mittelschrift_Regular9pt7b.h	31-7-2024 15:1
DIN_1451_Mittelschrift_Regular10pt7b.h	31-7-2024 15:1
DSEG7ClassicMini_Regular9pt7b.h	31-7-2024 15:1
DSEG7ClassicMini_Regular14pt7b.h	31-7-2024 15:1
DSEG7ClassicMini_Regular16pt7b.h	31-7-2024 15:1
DSEG7ClassicMini_Regular18pt7b.h	31-7-2024 15:1
DSEG7ClassicMini_Regular20pt7b.h	31-7-2024 15:1
DSEG7ClassicMini_Regular22pt7b.h	31-7-2024 15:1
DSEG7ClassicMini_Regular23pt7b.h	31-7-2024 15:1
DSEG7ClassicMini_Regular24pt7b.h	31-7-2024 15:1
DSEG7ClassicMini_Regular25pt7b.h	31-7-2024 15:1
DSEG7ClassicMini_Regular26pt7b.h	31-7-2024 15:1
FreeSans6pt7b.h	31-7-2024 15:1
FreeSans7pt7b.h	31-7-2024 15:1
FreeSans8pt7b.h	31-7-2024 15:1
Roboto_Regular7pt7b.h	31-7-2024 15:1
Roboto_Regular8pt7b.h	31-7-2024 15:1
Roboto_Regular9pt7b.h	31-7-2024 15:1
Roboto_Regular10pt7b.h	31-7-2024 15:1
Roboto_Regular11pt7b.h	31-7-2024 15:1

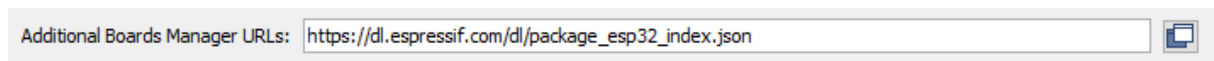
Start the Arduino IDE as shown above and open the .INO file in its new location.



```
CB_PLL_8 | Arduino 1.8.19
File Edit Sketch Tools Help
Upload
CB_PLL_8 7seg_31_19.h 7sega.h DIN_1451_Mittelschrift-Regular10pt7b.h DIN_1451_h
1 // CB PLL
2 // Jef Collin
3 // V 1.0 2024
4
5 // todo
6
7
8
9 #include <Wire.h>
10 #include <Adafruit_GFX.h>
11 #include <Adafruit_SSD1306.h>
12 #include <SPIFFS.h>
13 #include "Adafruit_MCP23X17.h"
14
```

We now have to install extra libraries that are not installed by default such as the ESP32 board libraries and Adafruit driver libraries (those listed in the #include section).

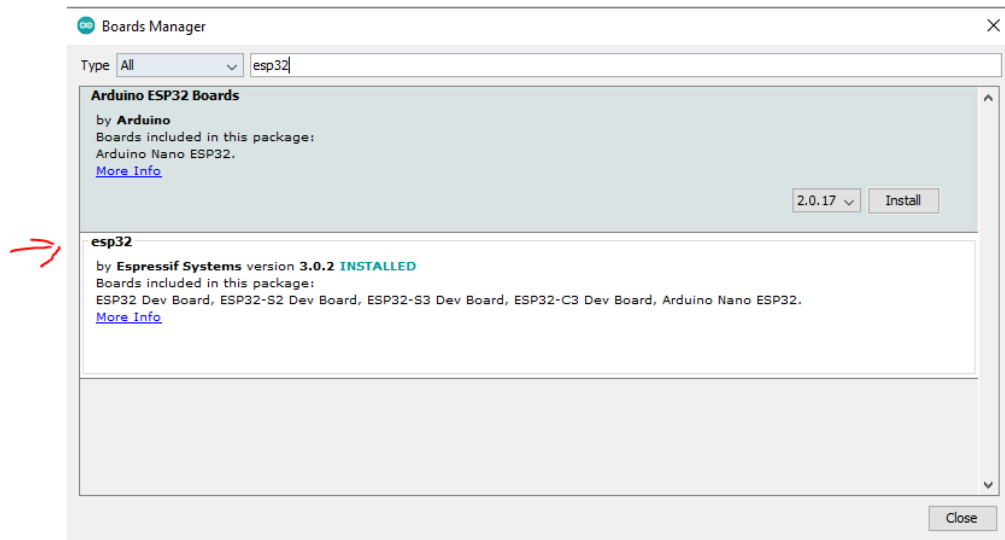
Go to *File->Preferences*, edit *Additional Boards Manager URLs* as follows:



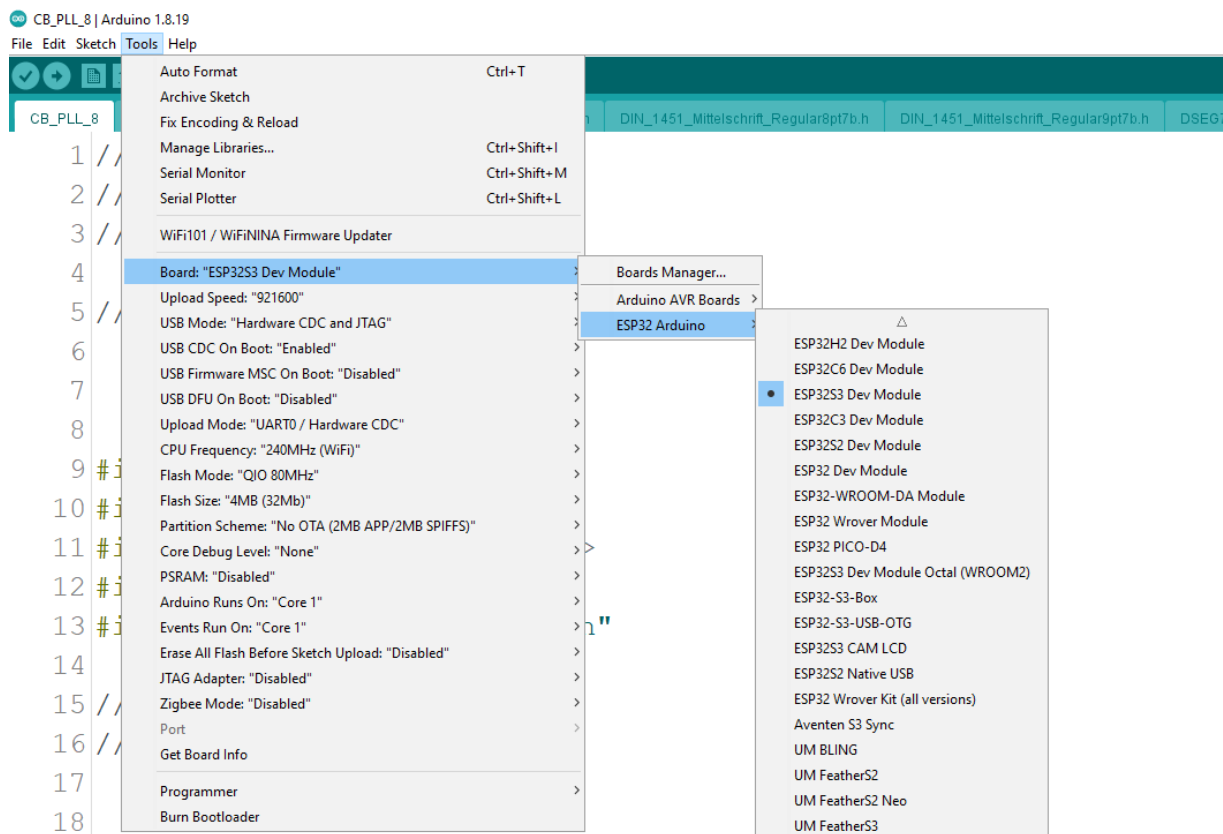
Important: to ensure the development environment is setup with the library versions which are known to work, install the version numbers listed. Do not update the libraries, ignore automatic messages from the IDE to update. When new versions of the source are released, re-check this document for the latest used library versions and update when required.

Go to *Tools->Board:...->Board Manager*.

In the search field type “esp32”, install *esp32 by Espressif Systems* version 3.0.2.



Go to *Tools-> Board:...->ESP32 Arduino* and select *ESP32S3 Dev Module*.

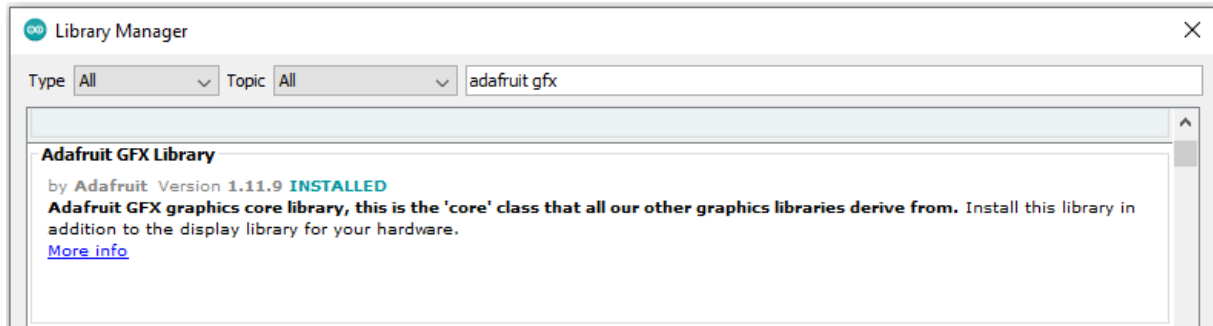


Verify the settings below *Board: “ESP32 Dev Module”*.

Go to *Tools-> Manage Libraries*, wait until the list is loaded, note that the library manager is very slow so be patient.

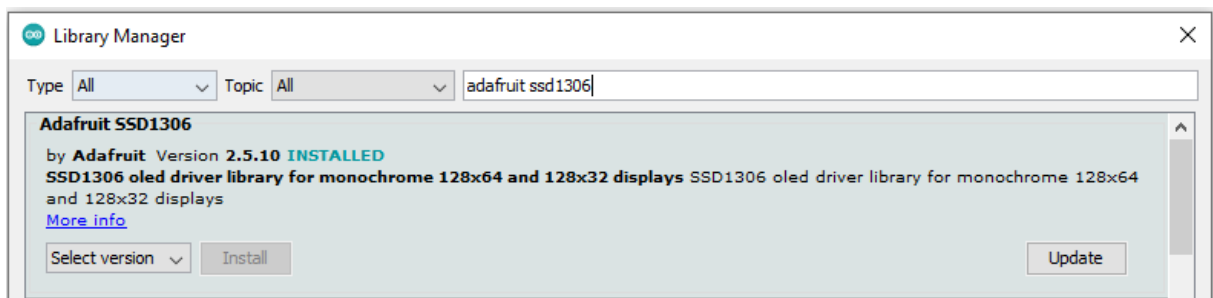
In the search field type “adafruit gfx”.

Install this library version 1.11.9, this installs the graphics system.



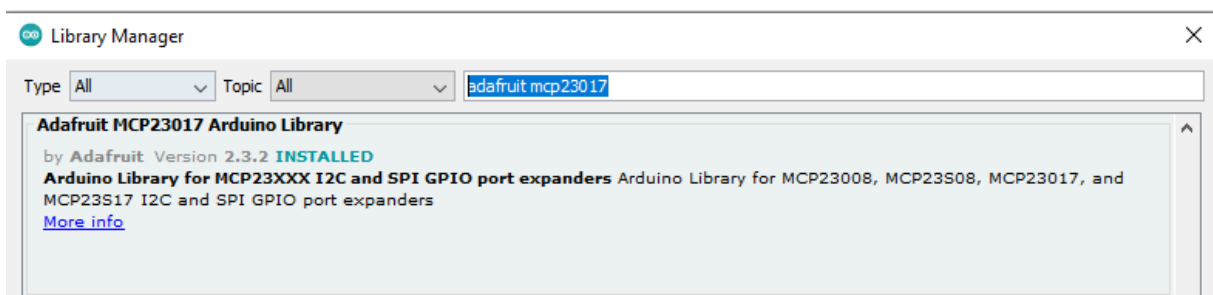
In the search field type “adafruit ssd1306”.

Install this library version 2.5.10, this installs the driver for the OLED display.

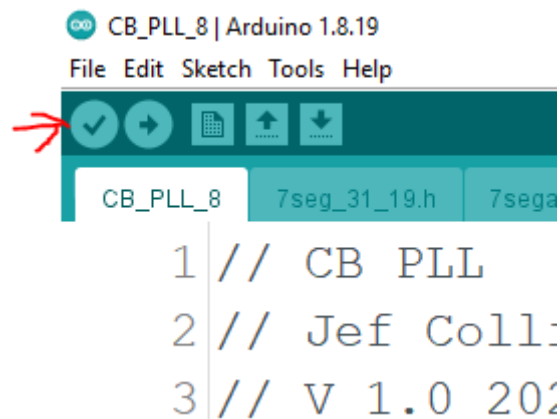


In the search field type “adafruit mcp23017”.

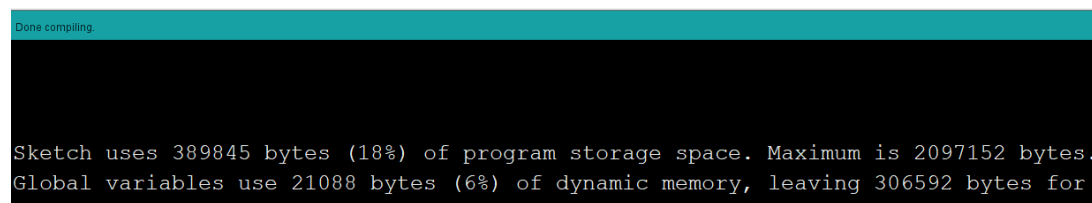
Install this library version 2.3.2, this installs the driver for the MCP23017 port expander.



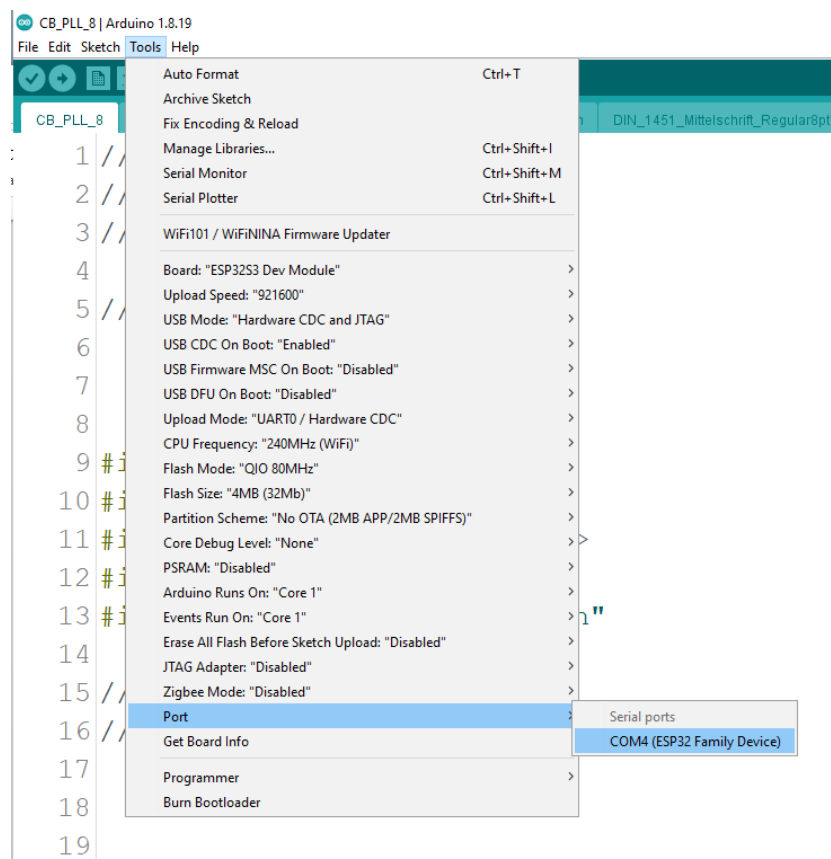
This completes the configuration, let's do a dry run first to verify the development environment, click the compile button.



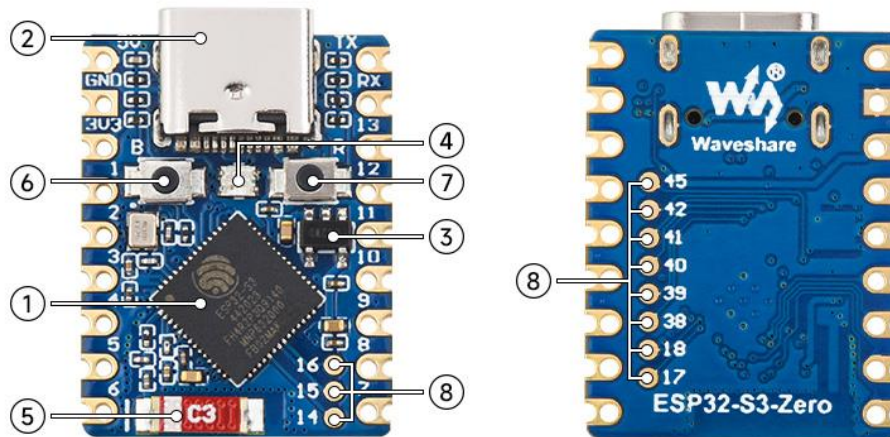
It should compile without errors.



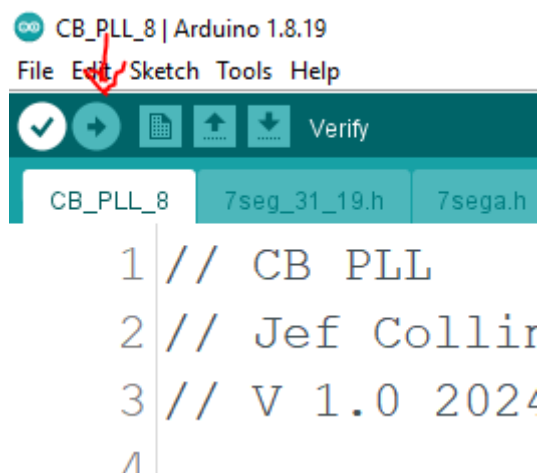
Connect the ESP32-S3 board with the USB C cable, Windows should detect it, go to *Tools->Port* and select the port assigned to it.



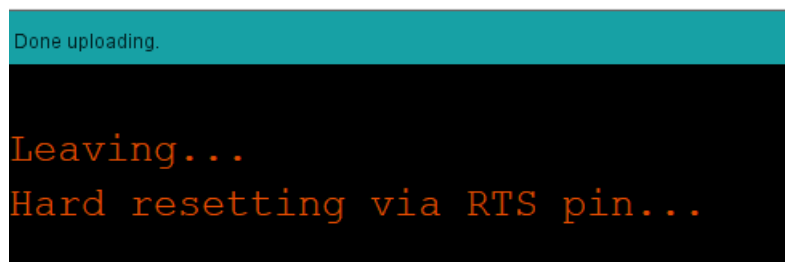
On the ESP32-S3 board, press both the Reset button marked R (nr 7) and the Boot button marked B (nr 6) , release the Reset button while holding the Boot button for a moment longer, this brings the board in download mode.



Click the download button.



It should compile and download without errors. Press the reset button on the ESP32-S3 to restart it (it will not be restarted automatically by the IDE).



This completed the programming.