Programmieren des ESP8266 (z.B. Wemos D1 mini) mit mBlock und der Arduino Software

BEESM, die Blockprogrammierumgebung, die du aus dem Workshop kennst, können wir leider (noch) nicht zur Nutzung privat zur Verfügung stellen. mBlock ist daher eine gute Alternative.

Auch mBlock ist eine App zur grafischen Programmierung mit Blöcken, die es Einsteigern/Innen erleichtert, eigene Arduino Programme zu schreiben. Anders als unsere eigene Software BEESM, unterstützt mBlock nicht die Programmierung der Controller wie z.B. den *Wemos D1 mini*, den wir in unseren Workshops benutzen. Daher müssen wir zum Hochladen des Codes zusätzlich die Arduino Software zu Hilfe nehmen.

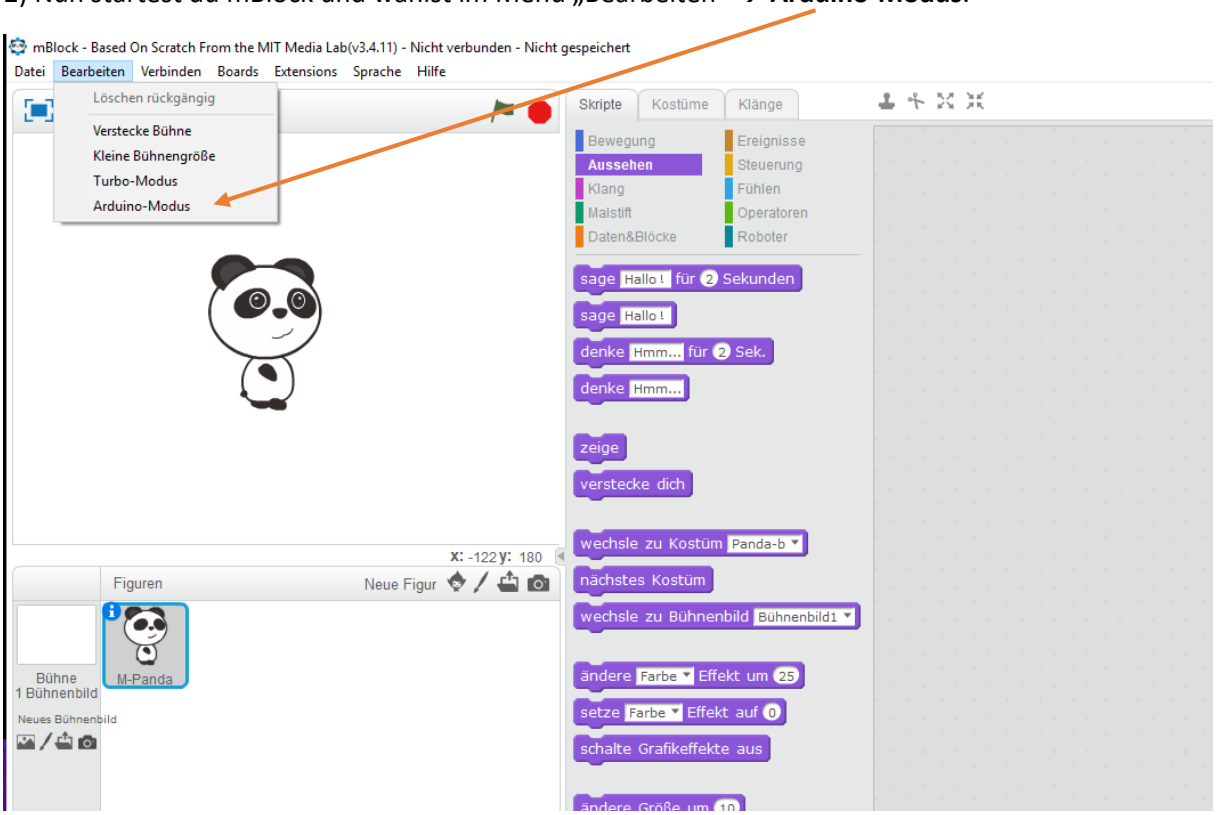
Bevor wir starten können, musst du also zwei Programme, mBlock und die Arduino IDE, auf dem Computer einrichten.

Einrichten von mBlock

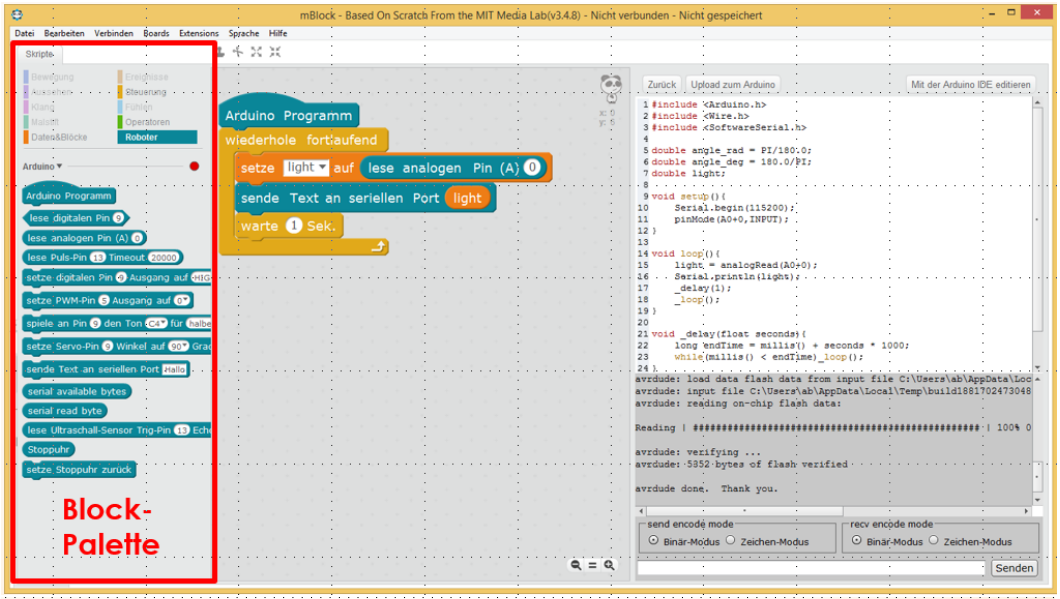
1) Download und installieren von mBlock: http://www.mblock.cc/?noredirect=en_US

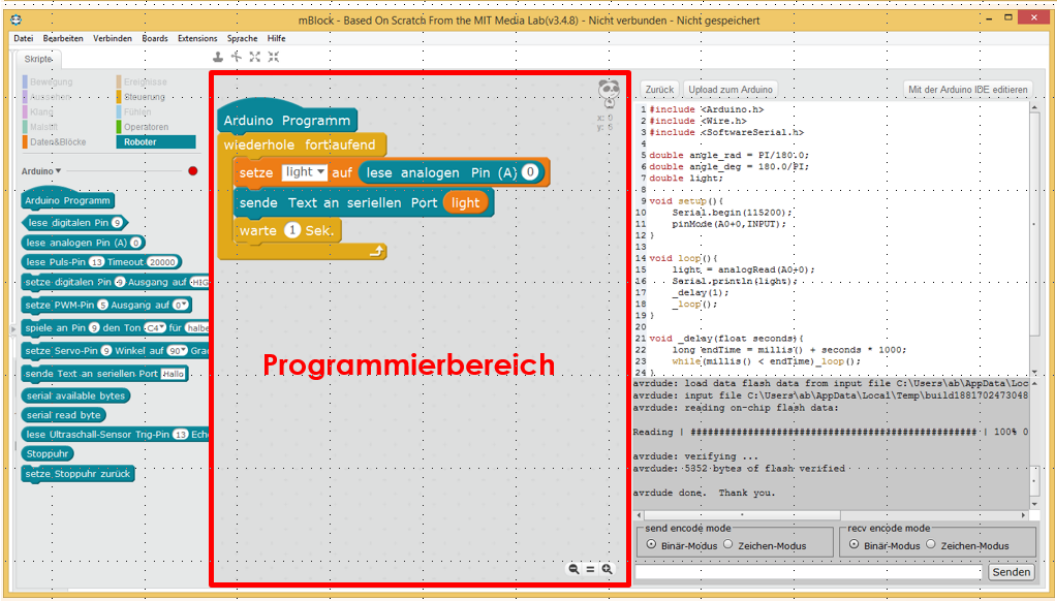
Version 3 hat sich bewährt!

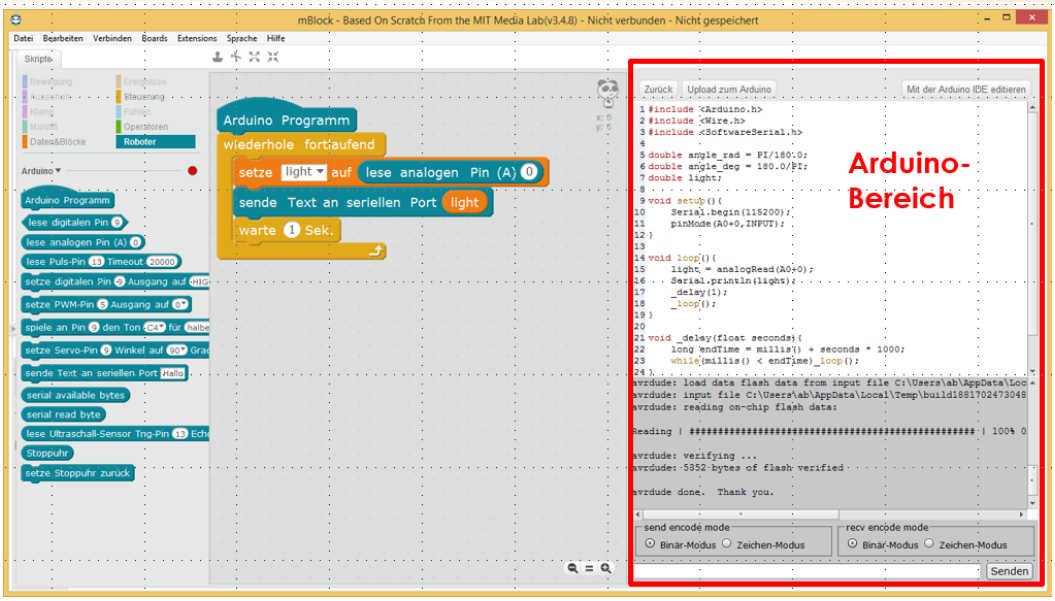
2) Nun startest du mBlock und wählst im Menü „Bearbeiten“ → **Arduino-Modus**.



3) mBlock ist wie folgt aufgebaut:



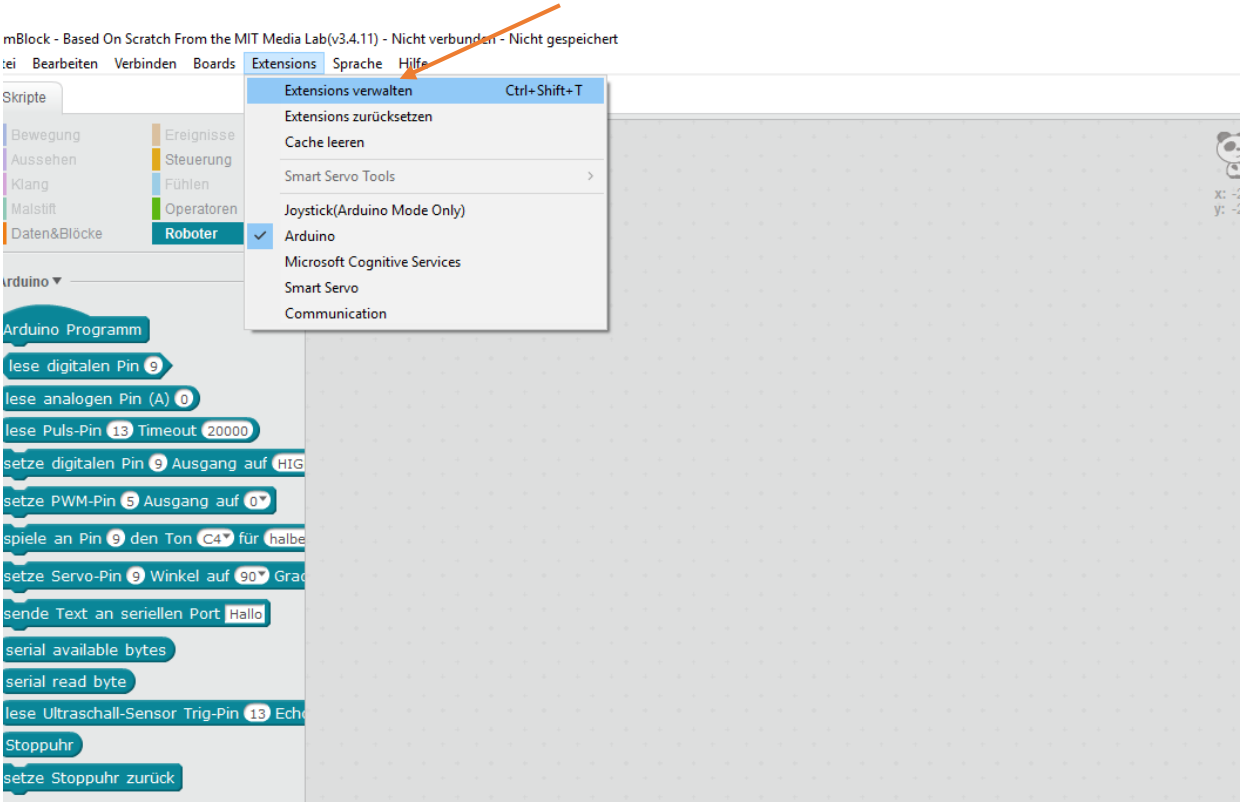




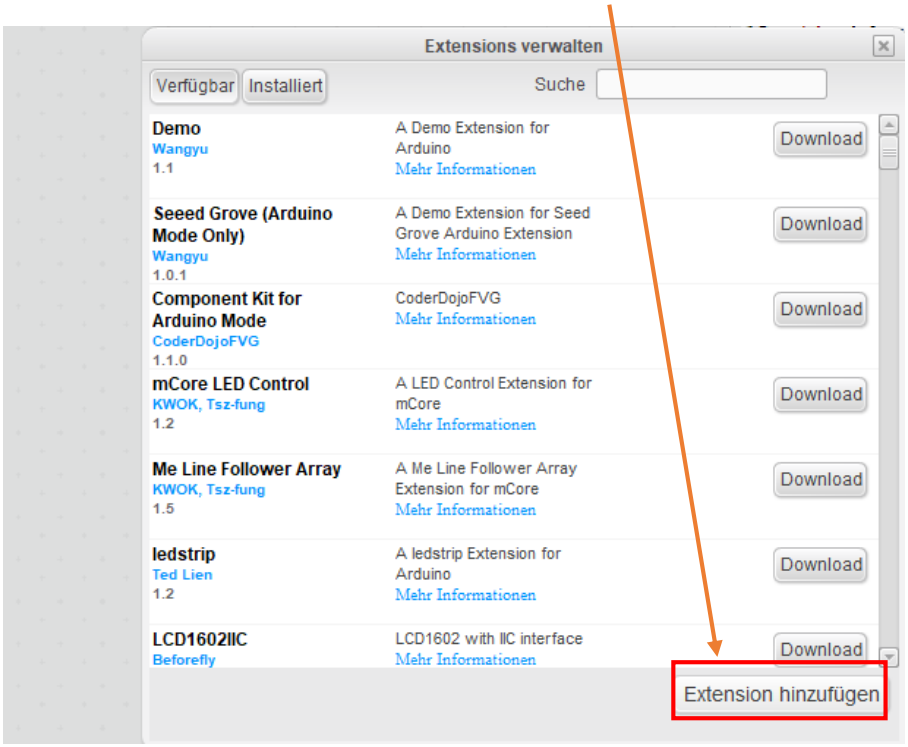
Die wichtigsten Blöcke findest du in der Blockpalette unter „Roboter“. Dort fehlen noch die passenden Blöcke, um unsere LEDs (auch Neopixel genannt) zu programmieren. Diese werden wie folgt eingebunden:

4) Extensions einbinden

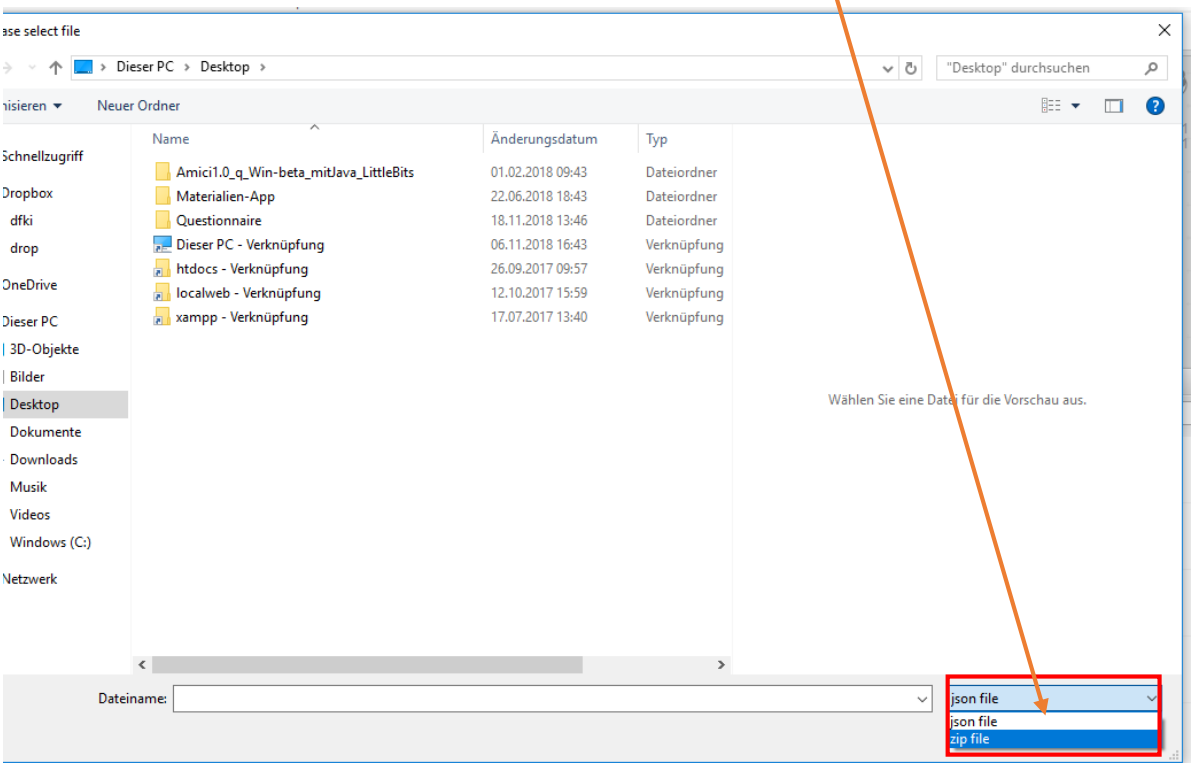
Im Menü über „Extensions“ → **Extensions verwalten** auswählen.



Im folgenden Fenster unten rechts „Extension hinzufügen“ auswählen.



Im folgenden Auswahlfenster unten rechts von „json file“ auf „zip file“ wechseln. Dann die .zip Datei „Neopixel Blöcke Wemos“, die auf deinem Computer gespeichert sein sollte, suchen und auswählen. Mit Öffnen unten rechts bestätigen. (Möchtest du mit dem Arduino (Uno) programmieren, dann wähle die .zip Datei „Neopixel Blöcke Arduino“ aus.)



Folgende Blöcke sollten jetzt bei mBlock in der Blockpalette unter Roboter zusätzlich auftauchen.

Setup: Wir definieren den Pin, an dem unsere LED am Arduino angeschlossen ist und wieviel LEDs wir in Reihe geschaltet haben

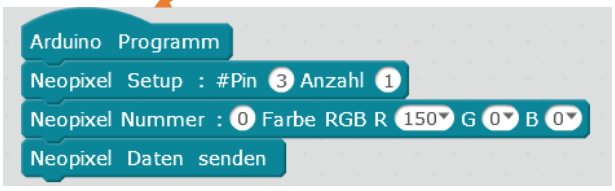
Farbe setzen: Das wievielte Licht in der Reihe soll gesetzt werden und welche Werte zwischen 0 …255 möchte jeweils ich für rot, grün, blau

Die oben gesetzten Farben muss ich nun an die Lichter senden. Sonst passiert nichts.

Optional: Ich kann die Helligkeit der Lichter setzen. Ohne Angabe leuchten die Lichter mit voller Intensität, das ist gleich 255.

Ein Programm beginnt immer mit dem „Arduino Programm“ Block, der ganz oben in der Blockpalette steht. Ein Beispielprogramm: Die LED leuchtet rot.

Hast du einen Arduino Uno programmiert, kannst du diesen Code nun in den Controller hochladen.



Das Hochladen in einen Esp-Controller wie der Wemos D1 mini funktioniert leider nicht über mBlock. Deswegen müssen wir den Code in die Arduino Software kopieren und diese nutzen um den Code hochzuladen. Die Arduino IDE müssen wir zuvor auch noch einrichten.

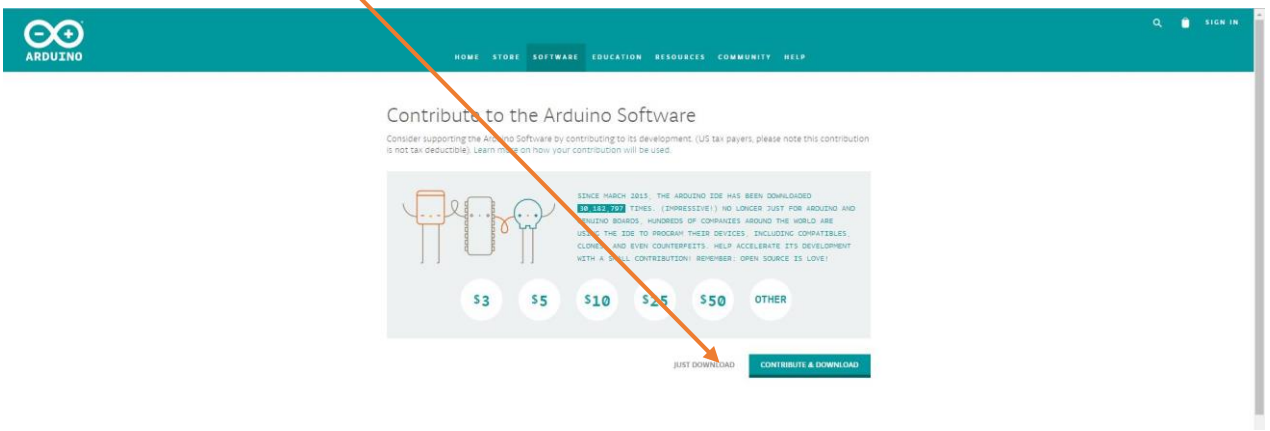
Einrichten der Arduino IDE

Installieren

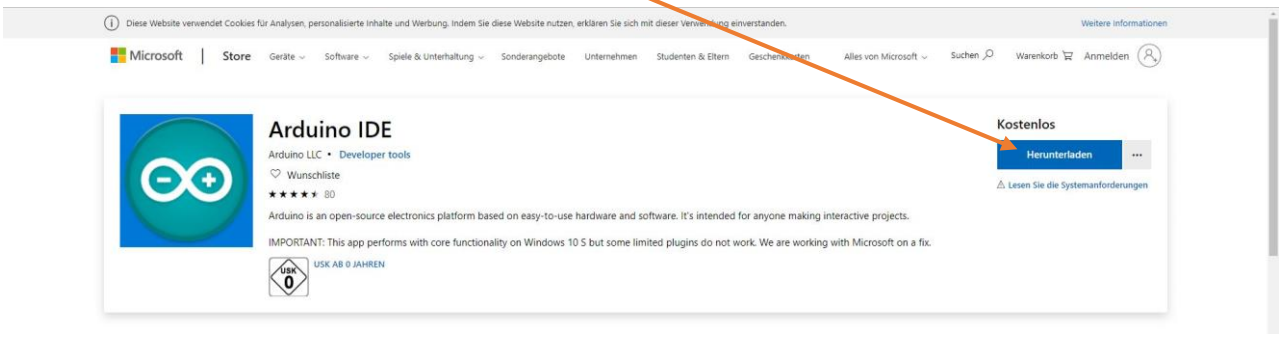
1) Öffne die Seite: https://www.arduino.cc/en/main/software

Wähle unter „Download the Arduino IDE“ dein Betriebssystem.
Hast du ein Windows System, empfehle ich Windows App auszuwählen.

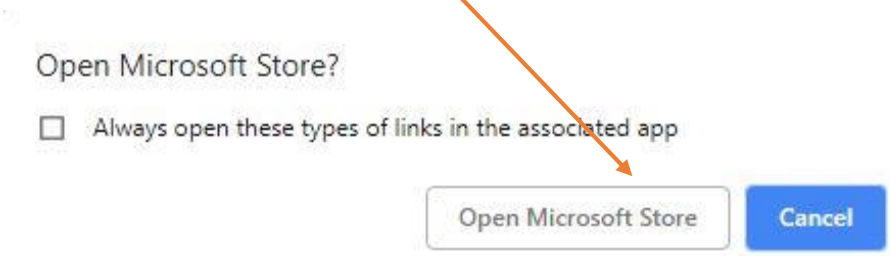
2) Wähle „JUST DOWNLOAD“



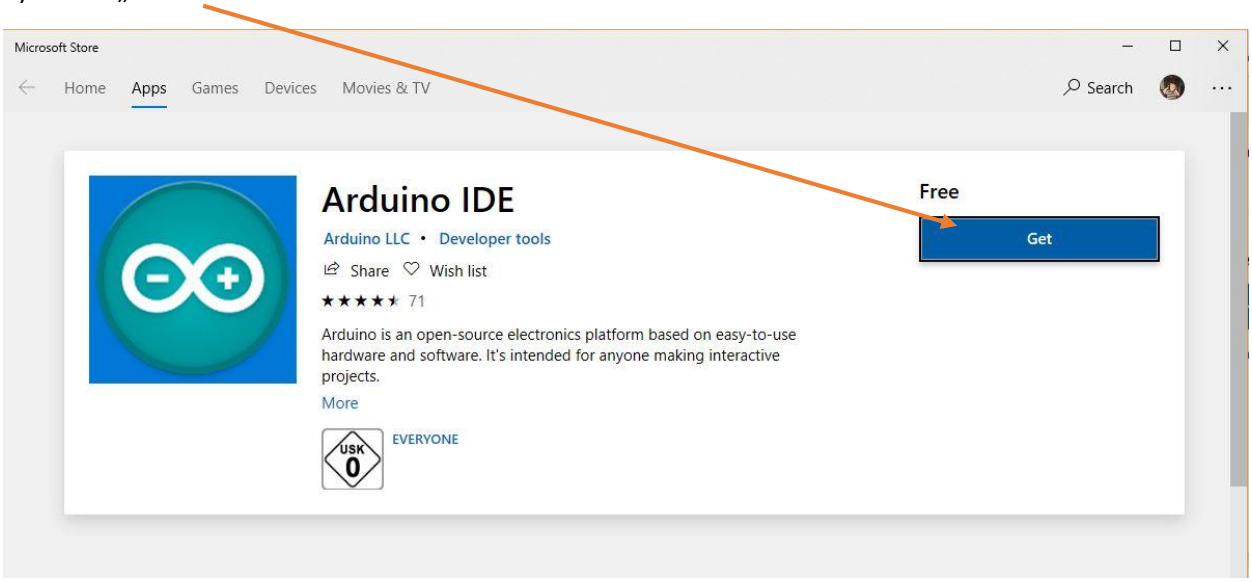
3) Wenn Du unter 1) „Windows app“ gewählt hast, öffnet sich ein Fenster, wie auf folgender Abbildung zu sehen. Wähle dort „Herunterladen“.
Ansonsten wird ein Programm zur Installation heruntergeladen, dass du im Anschluss starten must.



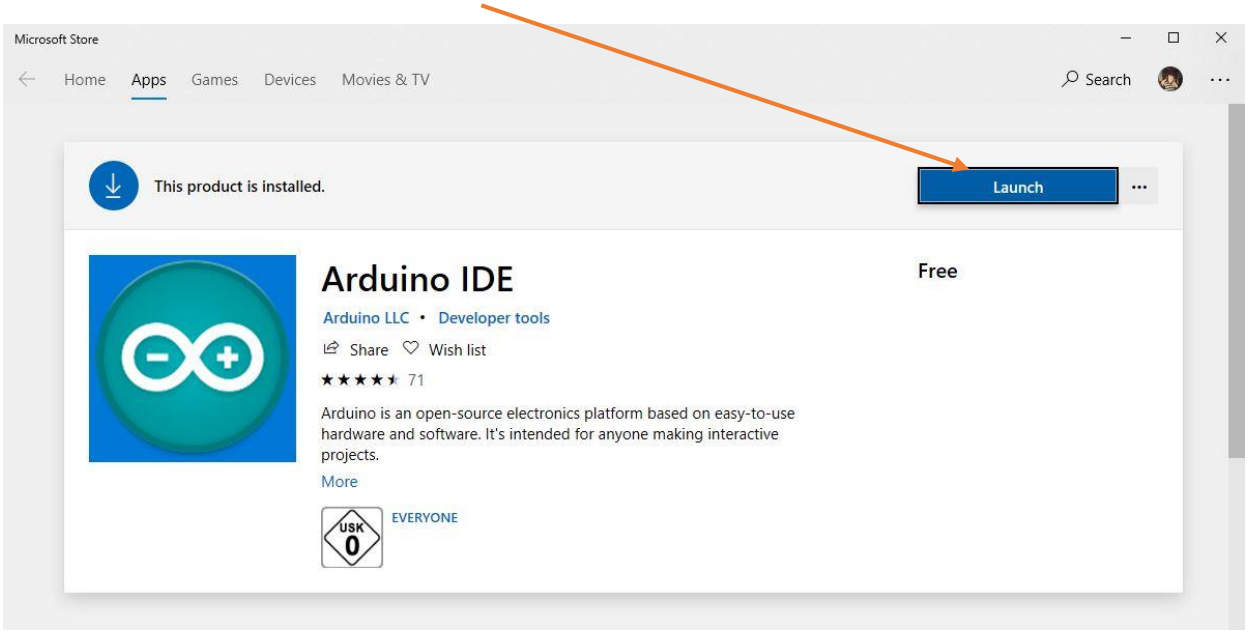
4) Falls sich dieses Fenster öffnet, wähle „Open Microsoft Store“, ansonsten weiter mit dem nächsten Schritt



5) Wähle „Get“

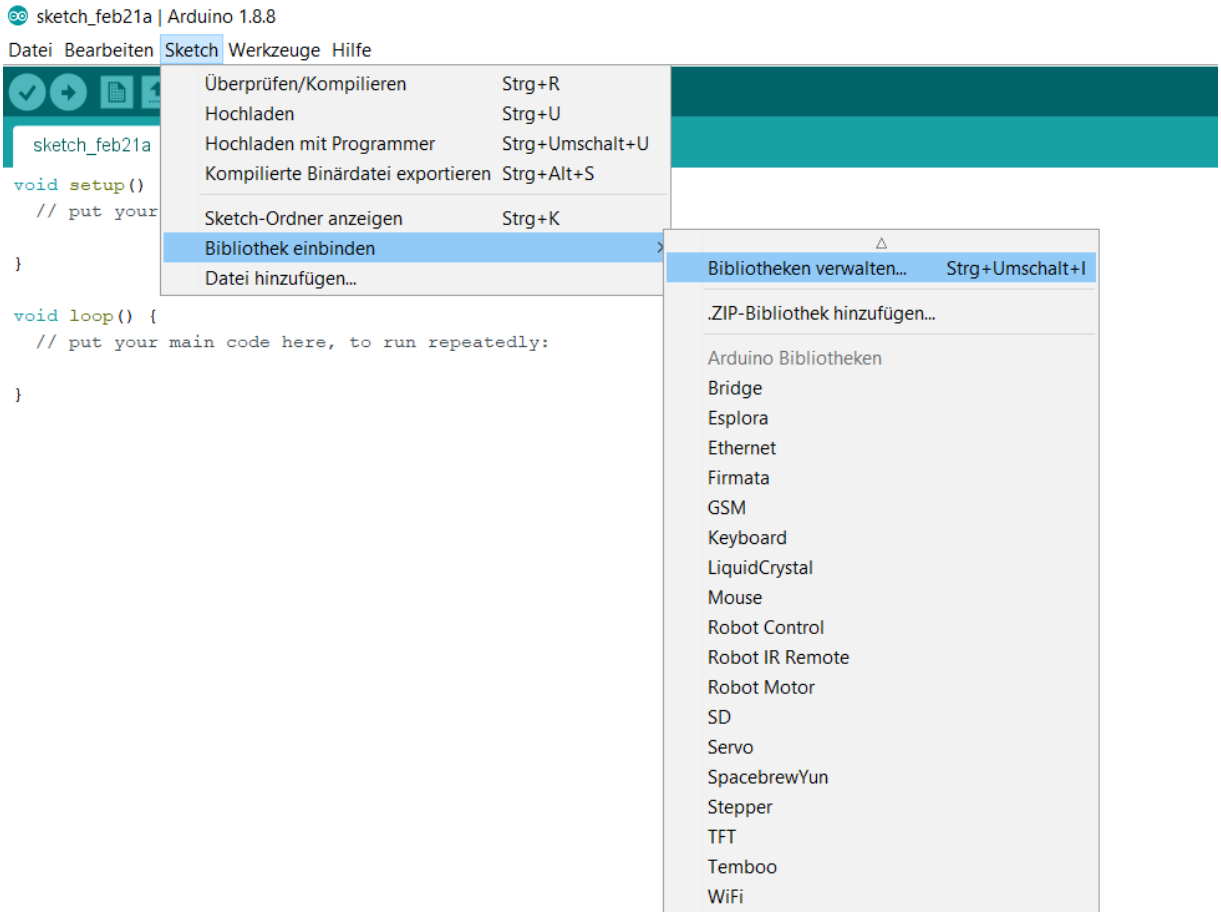


6) Wähle „Launch“ um die Arduino IDE zu starten

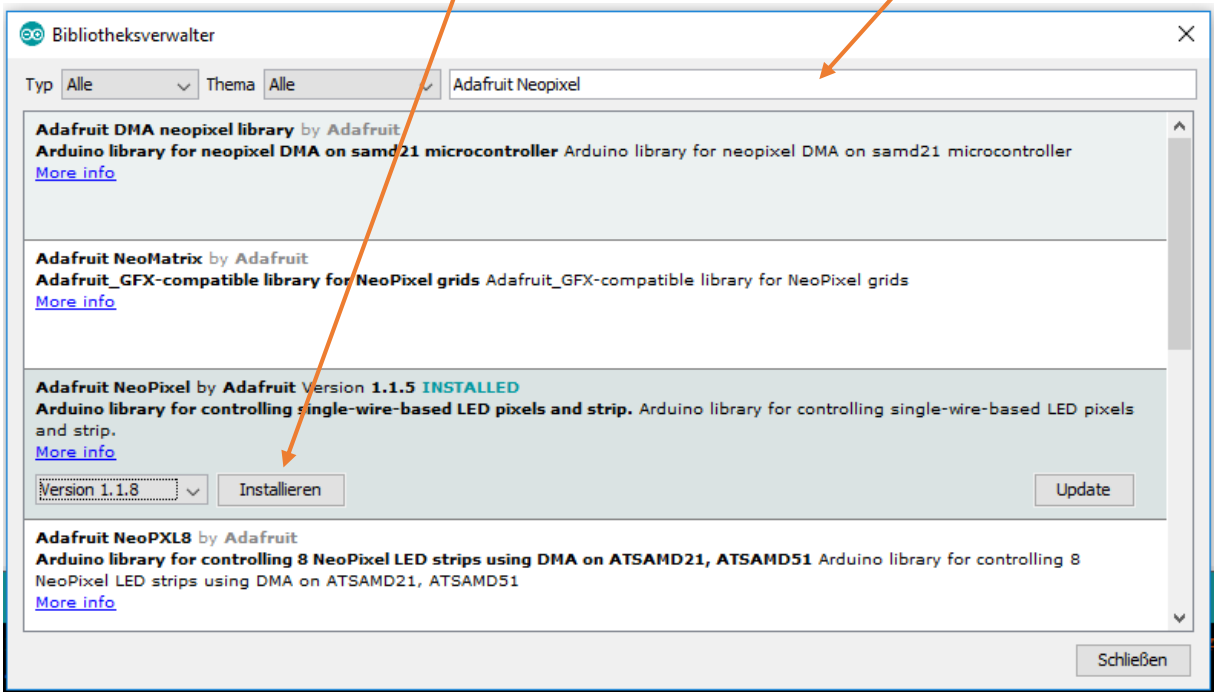


Bibliothek einfügen

1) Um eine Bibliothek einzufügen wählst du „Sketch“ → Bibliothek einbinden → **Bibliotheken verwalten**

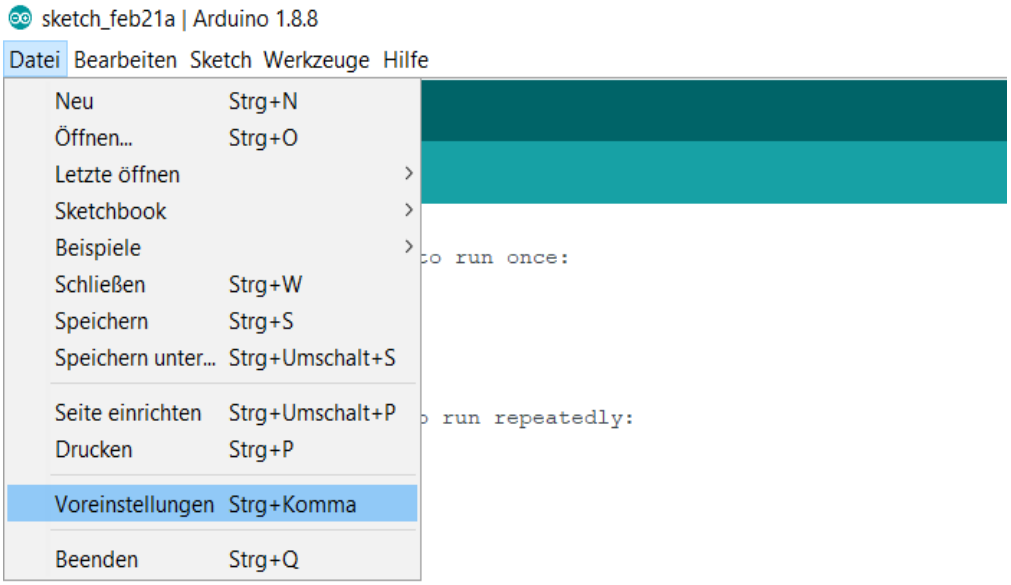


2) Im Fenster, welches sich daraufhin öffnet, trägst du „Adafruit Neopixel“ in die Suchleiste ein, klickst auf den Bereich „**Adafruit NeoPixel** by **Adafruit**“ und dann auf „Installieren“ („Installieren“ steht bei der ersten Installation unten rechts, wo auf der Abbildung „Update“ steht.)

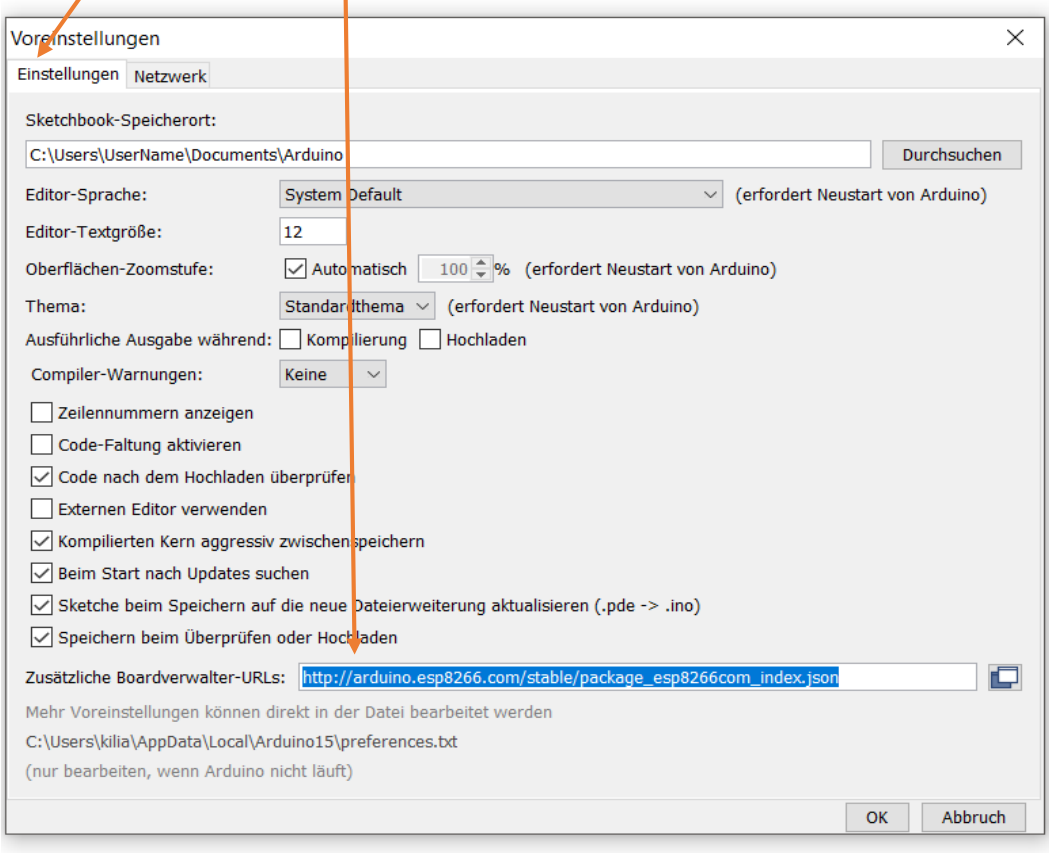


Den Wemos Controller einrichten

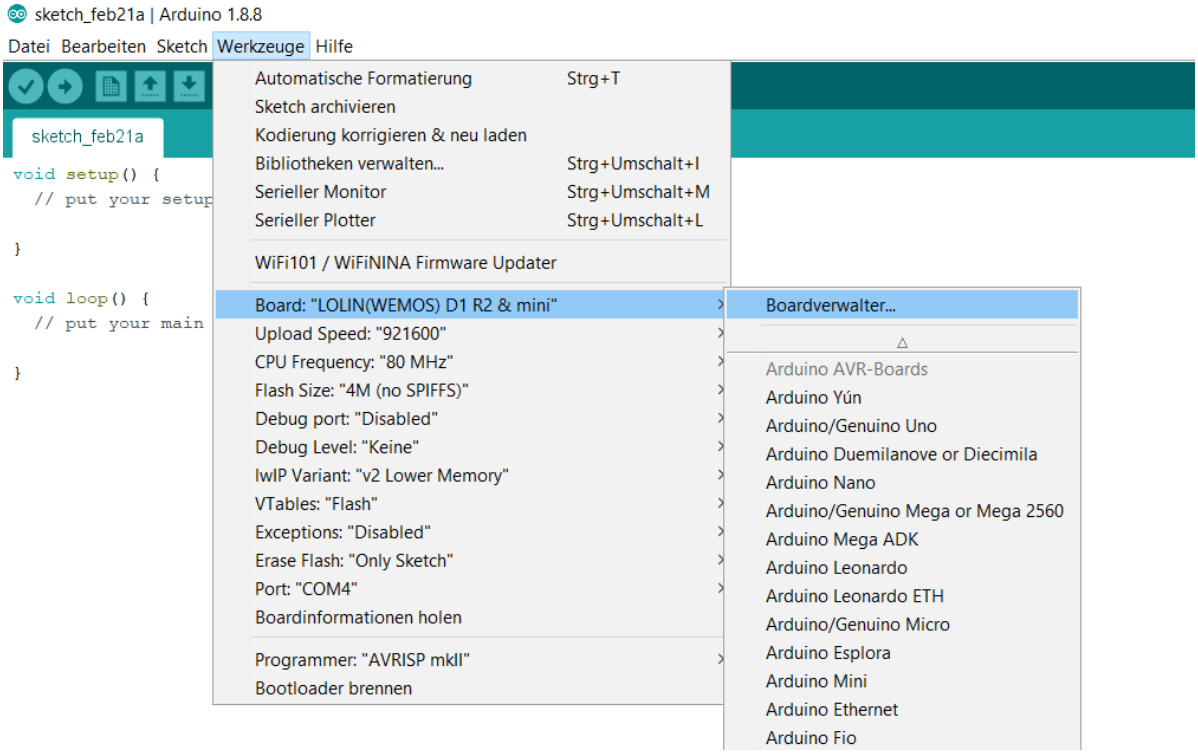
1) Öffne die Arduino IDE und wähle unter „Datei“ → **Voreinstellungen**



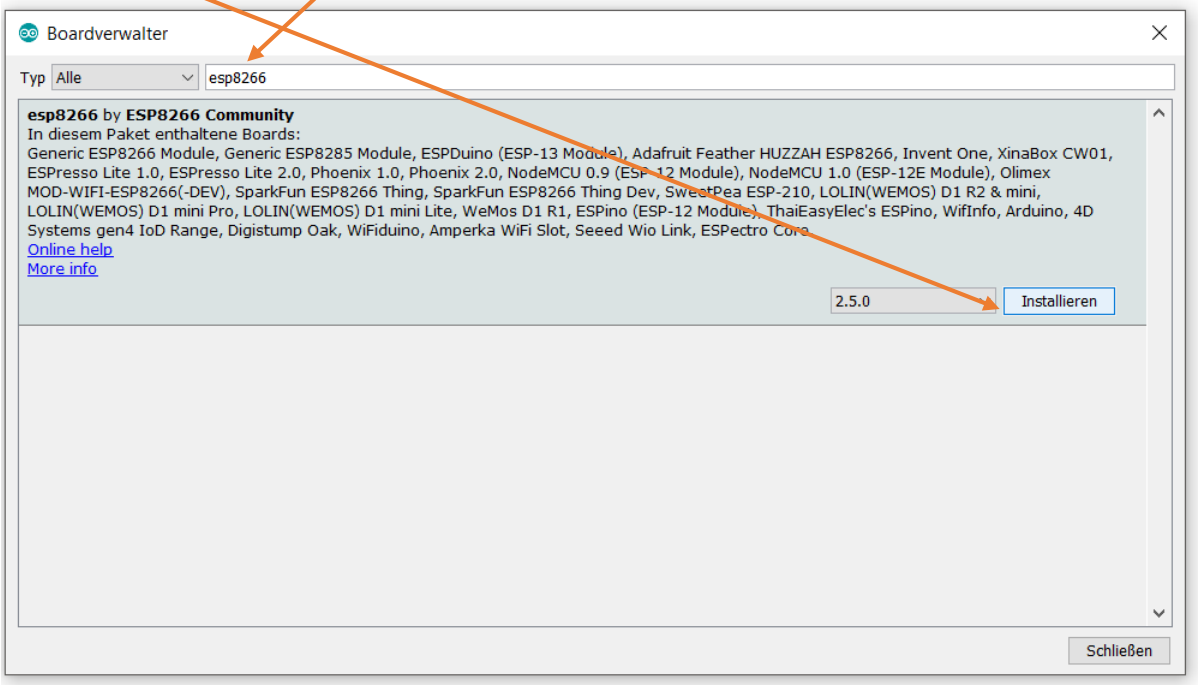
2) Unter „Einstellungen“ findest du das Feld „Zusätzliche Boardverwalter-URLs“, dort fügst du die folgende Adresse hinzu: http://arduino.esp8266.com/stable/package_esp8266com_index.json und bestätigst mit „OK“.



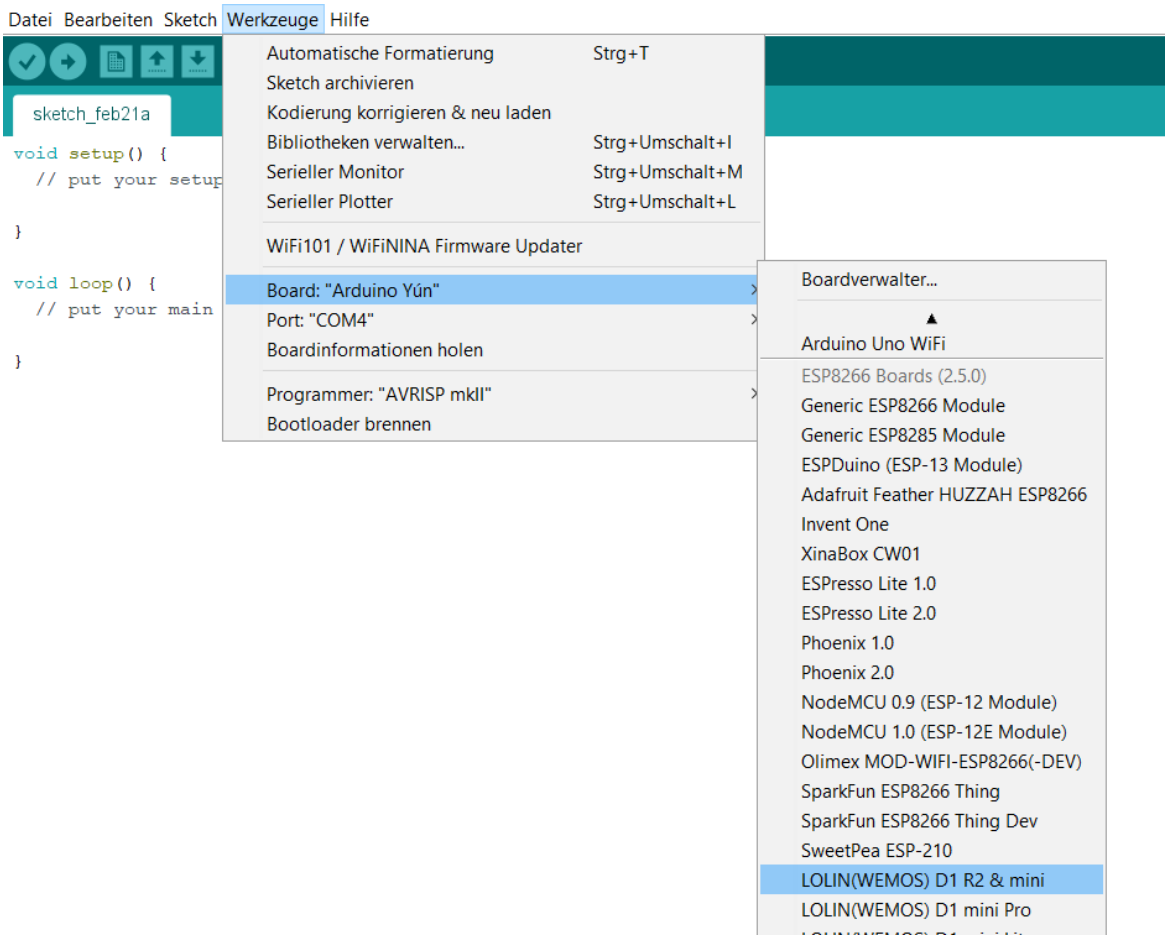
3) Im Menü „Werkzeuge“ wählst du das Untermenü „Board:…“ und dann den „Boardverwalter“ aus



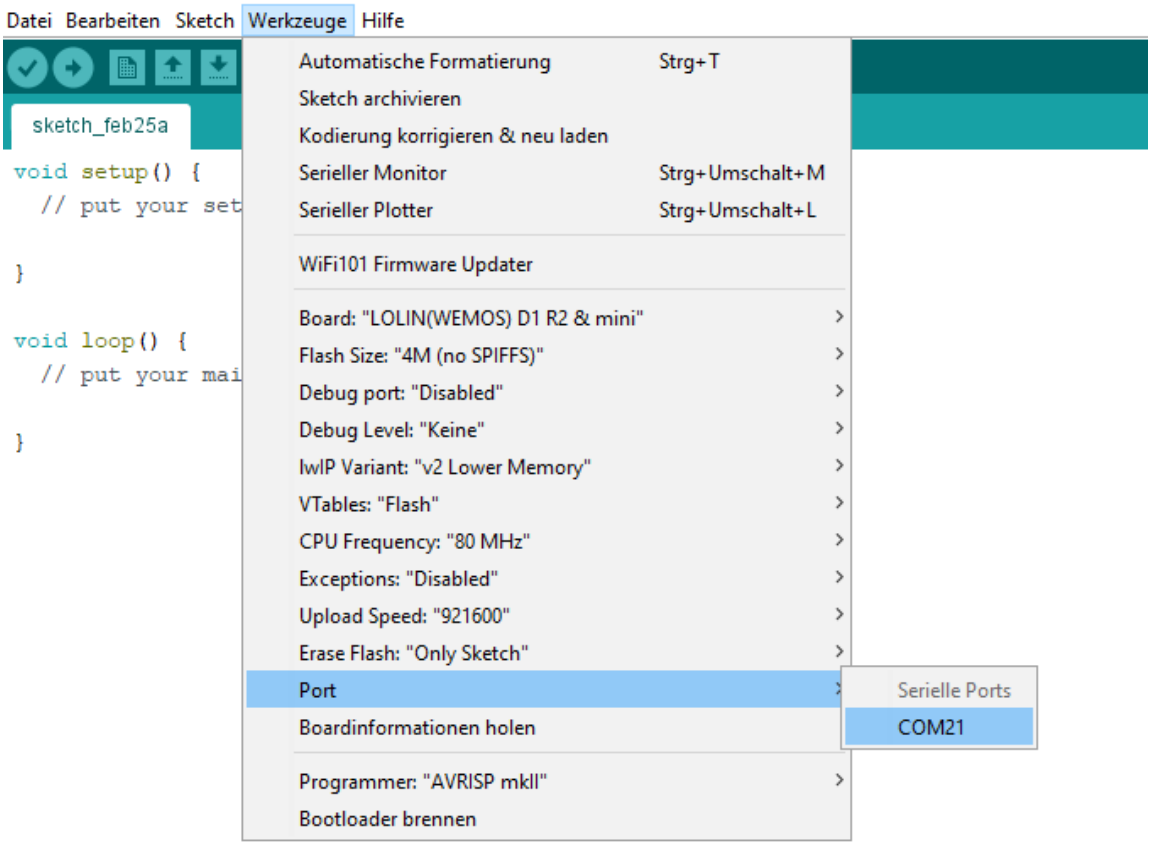
4) Nun gibst du „esp8266“ in das obere Suchfeld ein, wählst die aktuellste Version (hier 2.5.0) und klickst auf „Installieren“



5) Schließe den Mikrocontroller mit dem USB-Kabel an den Computer an. Nun wählst du das Board unter „Werkzeuge“ → „Board:…“ → LOLIN(WEMOS) D1 R2 & mini (oder Wemos D1 R2 & mini) (dieses steht sehr weit unten in der Liste)



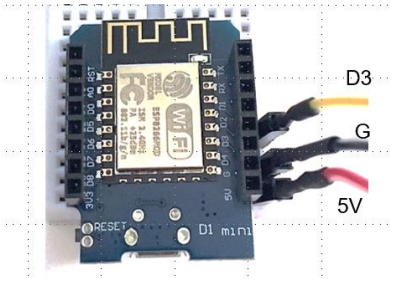
6) Abschließend musst du nur noch unter „Werkzeuge“ → „Port:…“ den richtigen Port auswählen. Unter Windows wird dieser immmer mit „COM“ und eine Zahl bezeichnet. Beachte, dass der Wemos Controller über USB mit deinem Rechner verbunden sein muss.

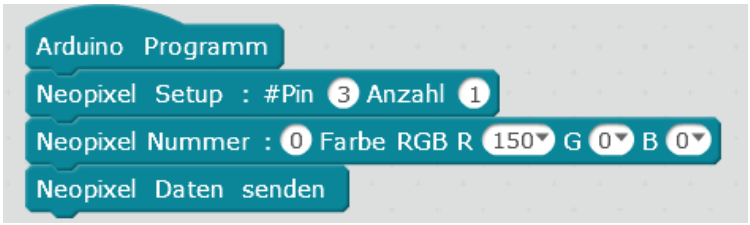


Die Vorarbeiten sind erledigt. Nun geht's los!
Neopixel programmieren mit mBlock und der Arduino IDE

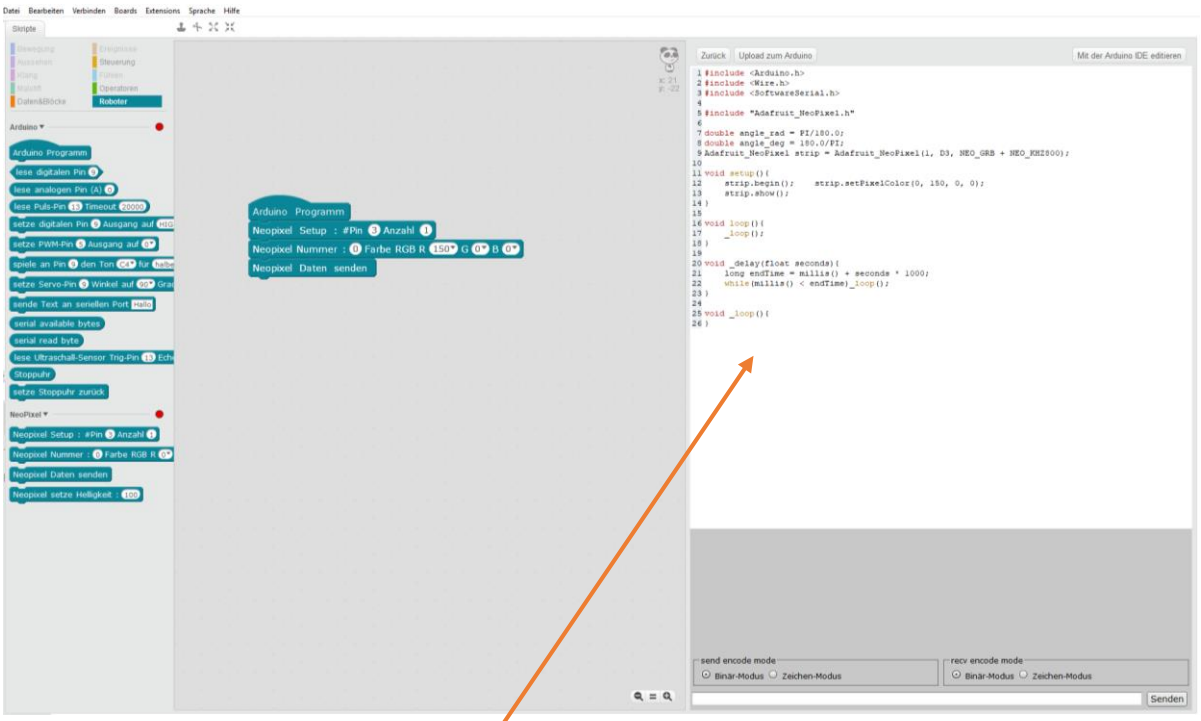
Zunächst gestalten wir das einfache Beispielprogramm von oben: Eine LED soll rot leuchten.

Das folgende Beispielprogramm lässt eine LED (Anzahl Pixel -> 1), die mit der Datenleitung (gelbes, grünes oder blaues Kabel) an Pin 3 (D3) des Controllers angeschlossen ist, rot leuchten.





In mBlock sieht dein Programm wie folgt aus:



Nun markieren wir den Code im Arduino Bereich ab der Zeile 5, also ab `#include "Adafruit_NeoPixel.h"`.

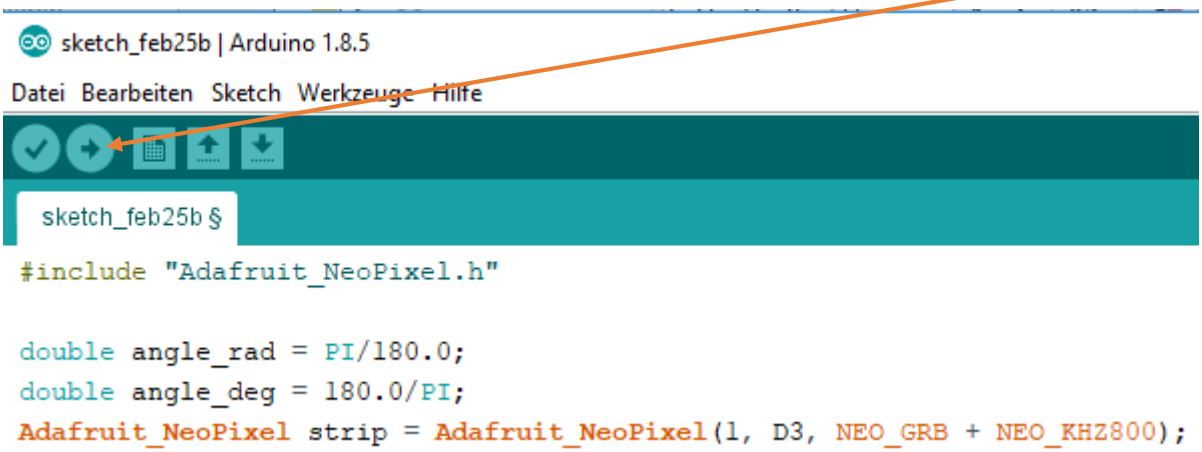
Dazu setzt du den Curor (⌶) an den Anfang der 5. Zeile kurz vor das `#include` und ziehst diesen mit gedrückter linker Maustaste runter bis dein Code wie in folgender Abbildung blau markiert ist.

Danach kopierst du den markierten Code. Dazu gibt es zwei Möglichkeiten:
- Du drückst auf deiner Tastatur die Tasten „strg“ und „c“ gleichzeitig, oder
- Du drückst die rechte Maustaste und wählst „kopieren“ aus dem Kontextmenü.

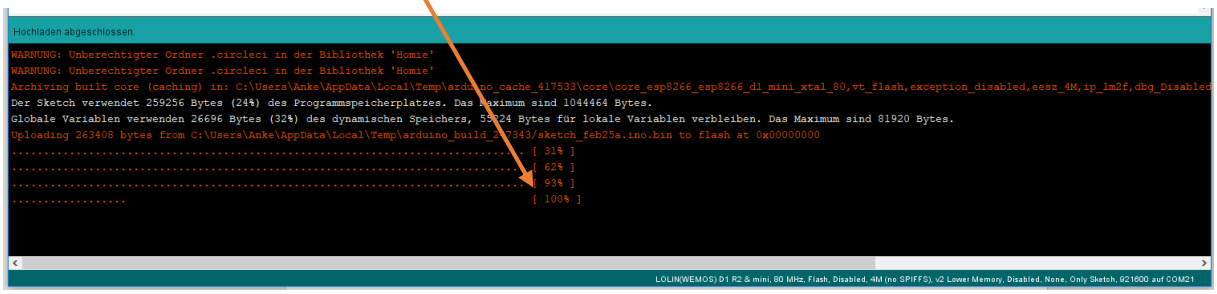
Danach startest du die Arduino IDE und wählst dort „Datei“ → **Neu**

Danach löschst du im Programmierbereich den bereits vorhandenen Code und fügst den kopierten Code aus mBlock ein. Dazu klickst du in den leeren Programmierbereich und drückst entweder die Tasten „strg“ und „v“ gleichzeitig auf deiner Tastatur, oder gehst über das Menü „Bearbeiten“ → Einfügen.

Danach kannst du den Code auf deinen Controller hochladen, indem du auf den rechten Pfeil klickst.



Beim Hochladen wird dein Programm zunächst kompiliert, das heißt in eine für den Computer lesbare Form gebracht, und danach auf deinen Controller übertragen. Das dauert eine Weile. Steht in dem schwarzen Status-Fenster unten „100%“, wie in der folgenden Abbildung, wurde dein Programm erfolgreich auf den Controller übertragen.



Wenn Fehlermeldungen auftauchen, dann überprüfe:
-Ist dein Controller über ein USB Kabel mit deinem Rechner verbunden?
-Ist dein Port ausgewählt (über „Werkzeuge“ → „Port“)?
-Ist der richtige Controller ausgewählt (Werkzeuge → Board: LOLIN(WEMOS) D1 R2 & mini)?

Stecke ggf. das usb-Kabel von deinem Computer ab und wieder an, wähle erneut den Port aus und lade dein Programm nochmal hoch.