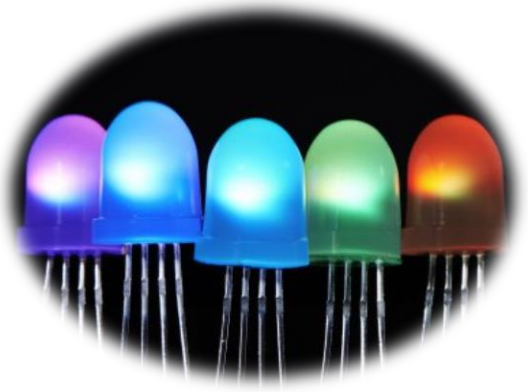
Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

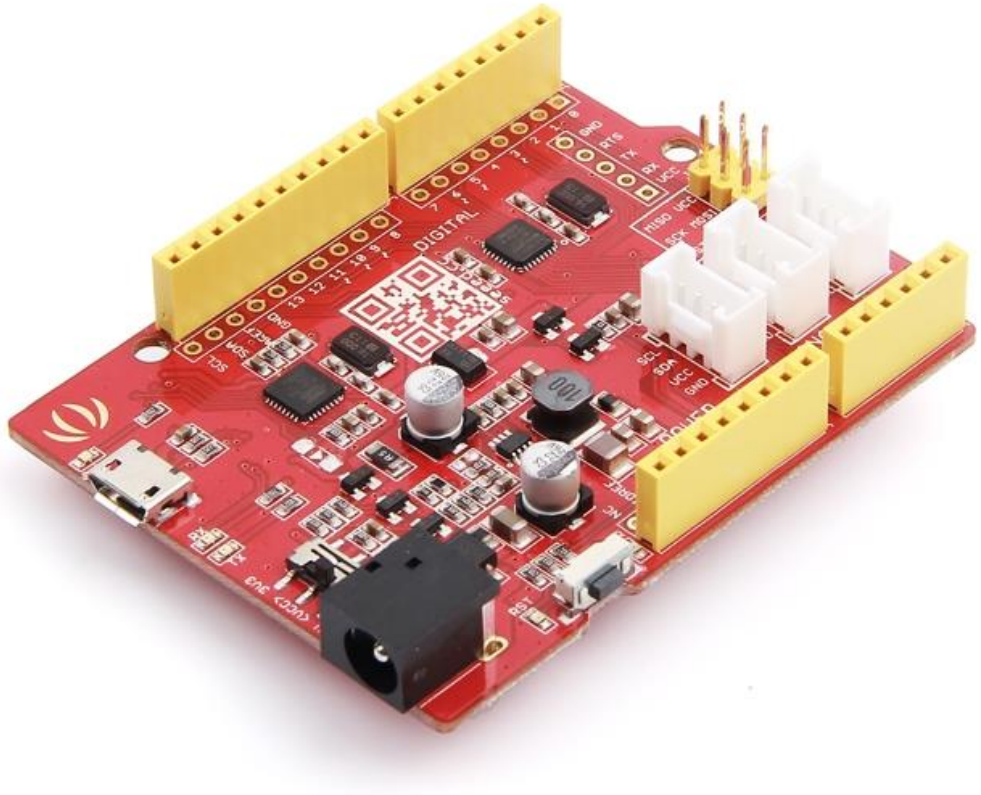
Smartes Stimmungslicht

smile IT

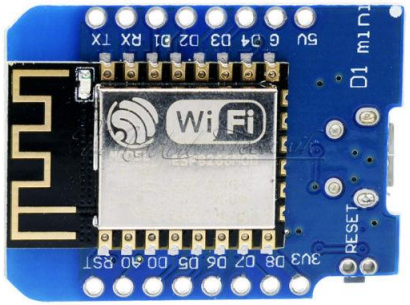
SMART · FUTURE · ME



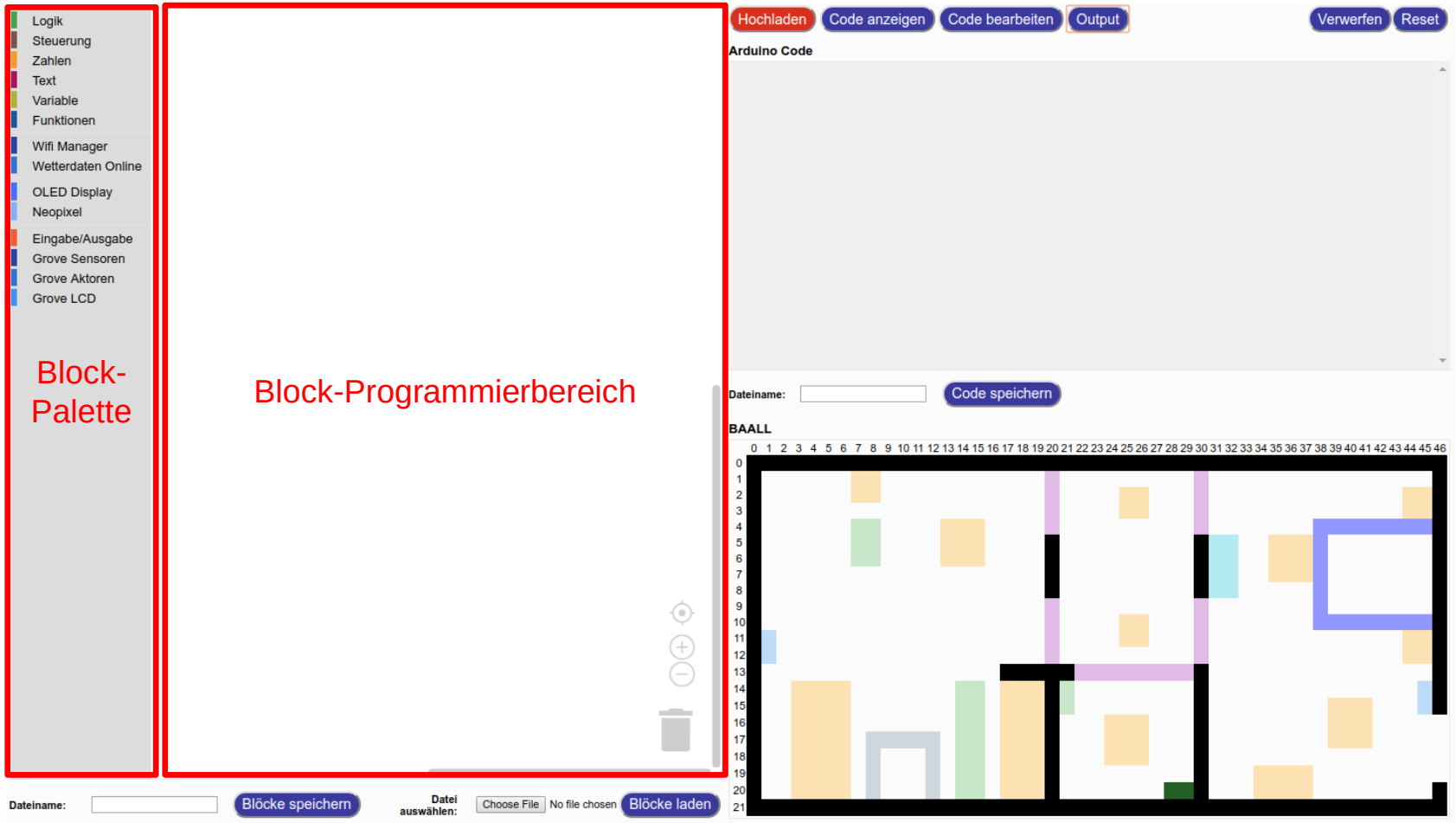
Hands-On Arduino



- Der Arduino ist ein Mini-Computer
- Mit dem Arduino lassen sich schnell und einfach Sensoren und LEDs oder Motoren ansteuern.

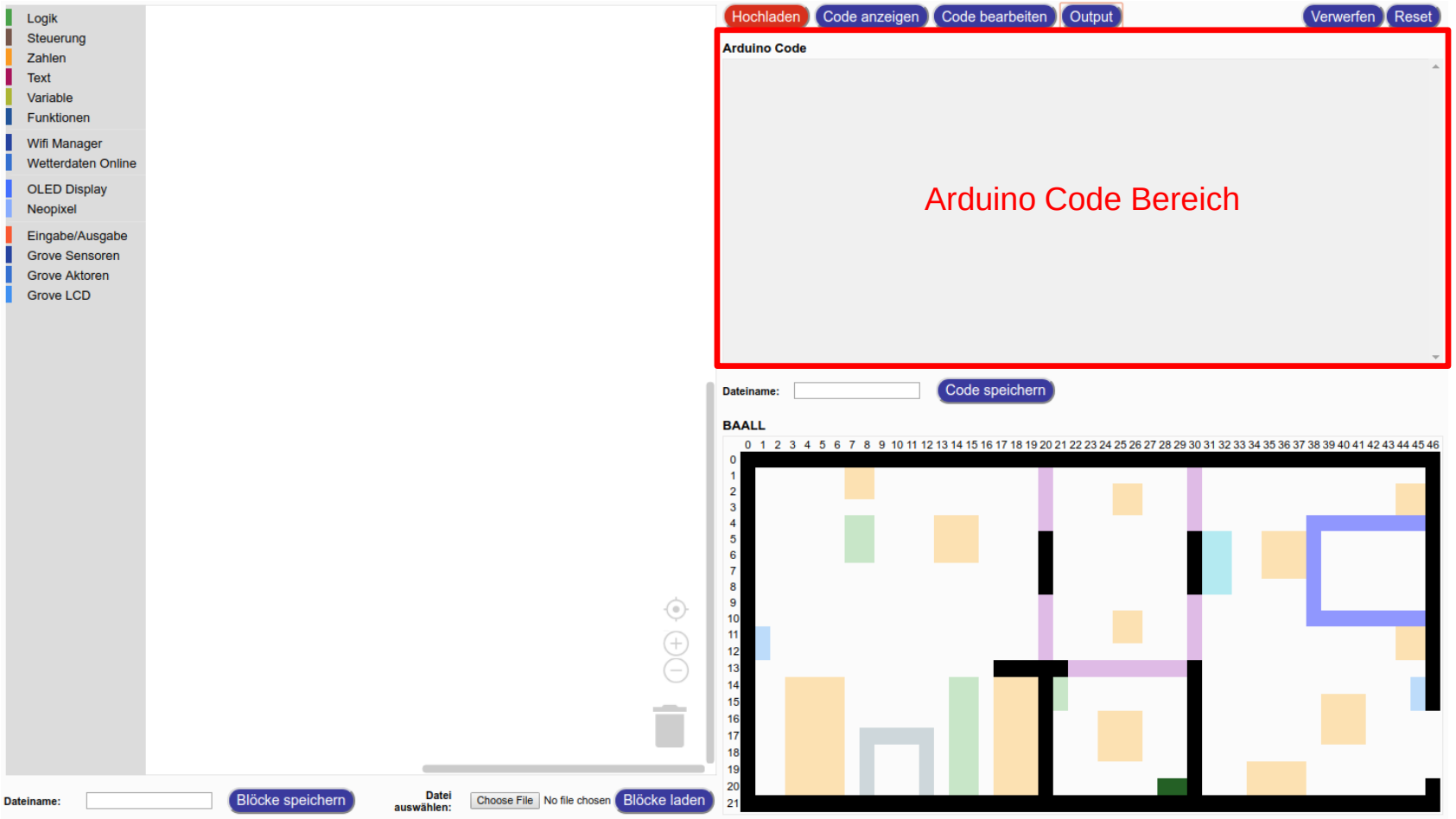


Die Entwicklungsumgebung: BEESM



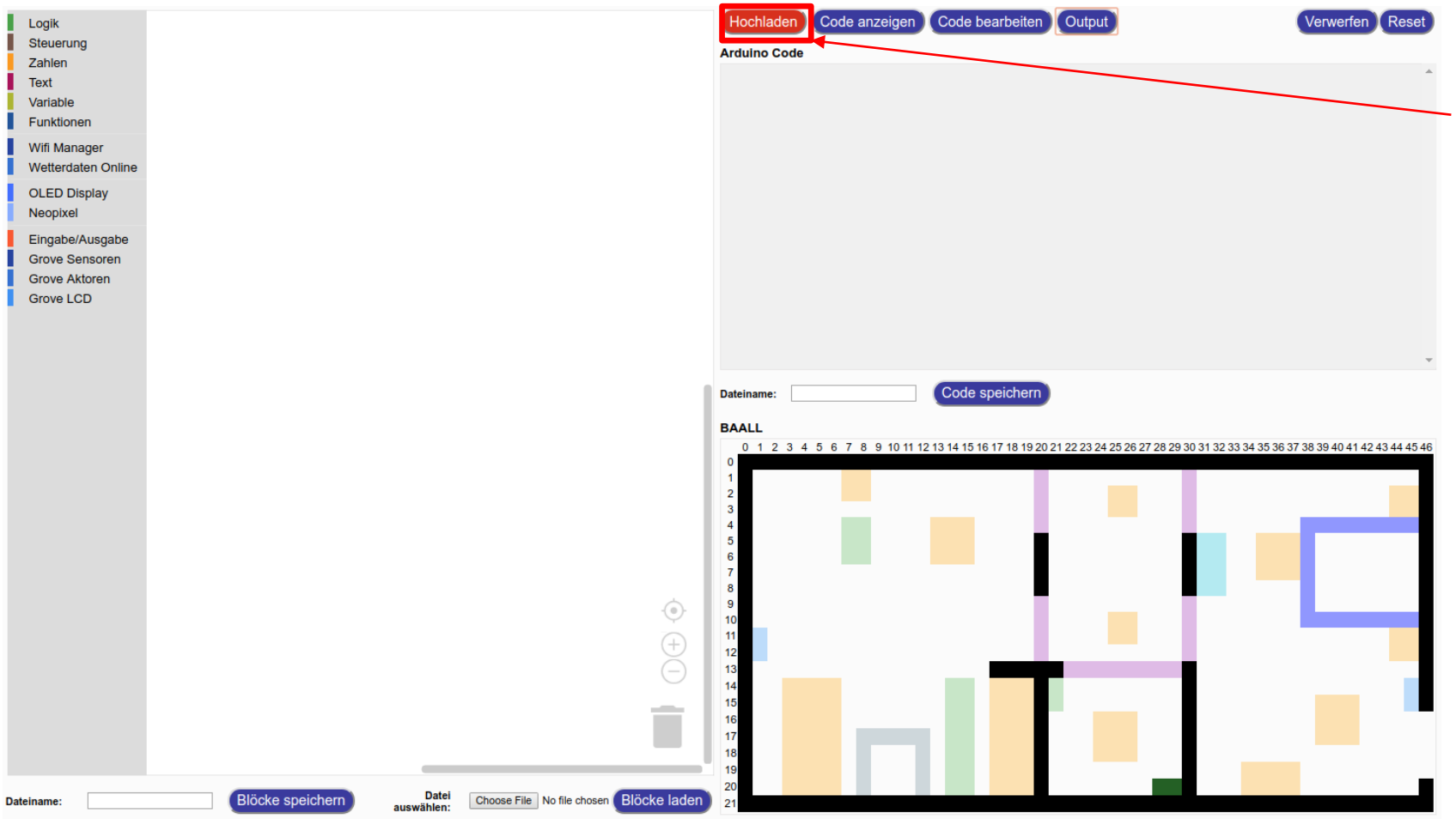
In der Blockpalette und im Programmierbereich arbeiten wir und stellen unser Programm aus verschiedenen Blöcken zusammen.

Die Entwicklungsumgebung: BEESM



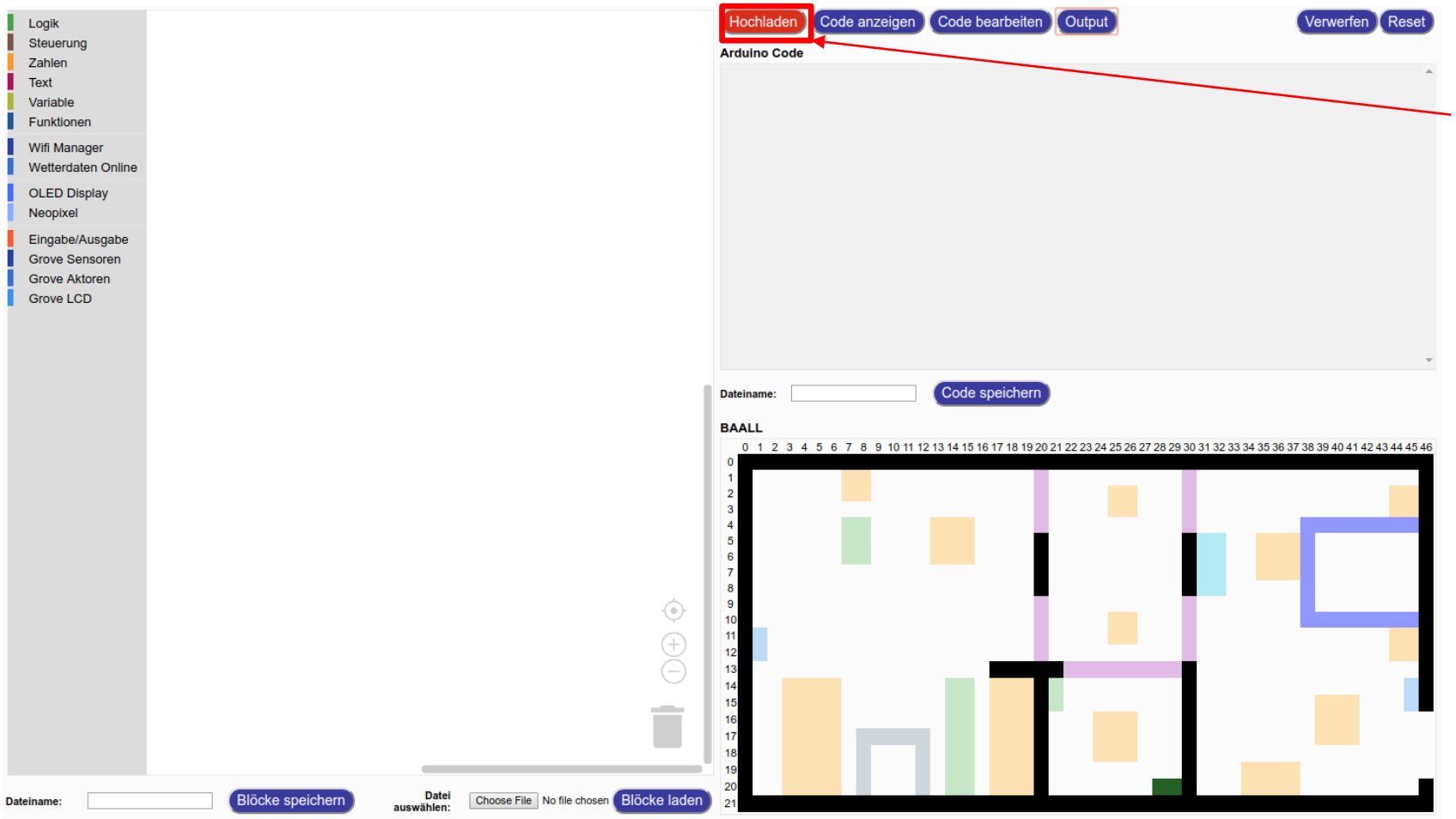
Im Code Bereich sehen wir den Code, der aus den Blöcken, die wir im Block-Programmierbereich zusammengestellt haben, generiert wird.

Die Entwicklungsumgebung: BEESM



Mit einem Klick auf Hochladen wird das Programm auf den Controller übertragen. Dies kann ein bisschen dauern. Also Geduld!

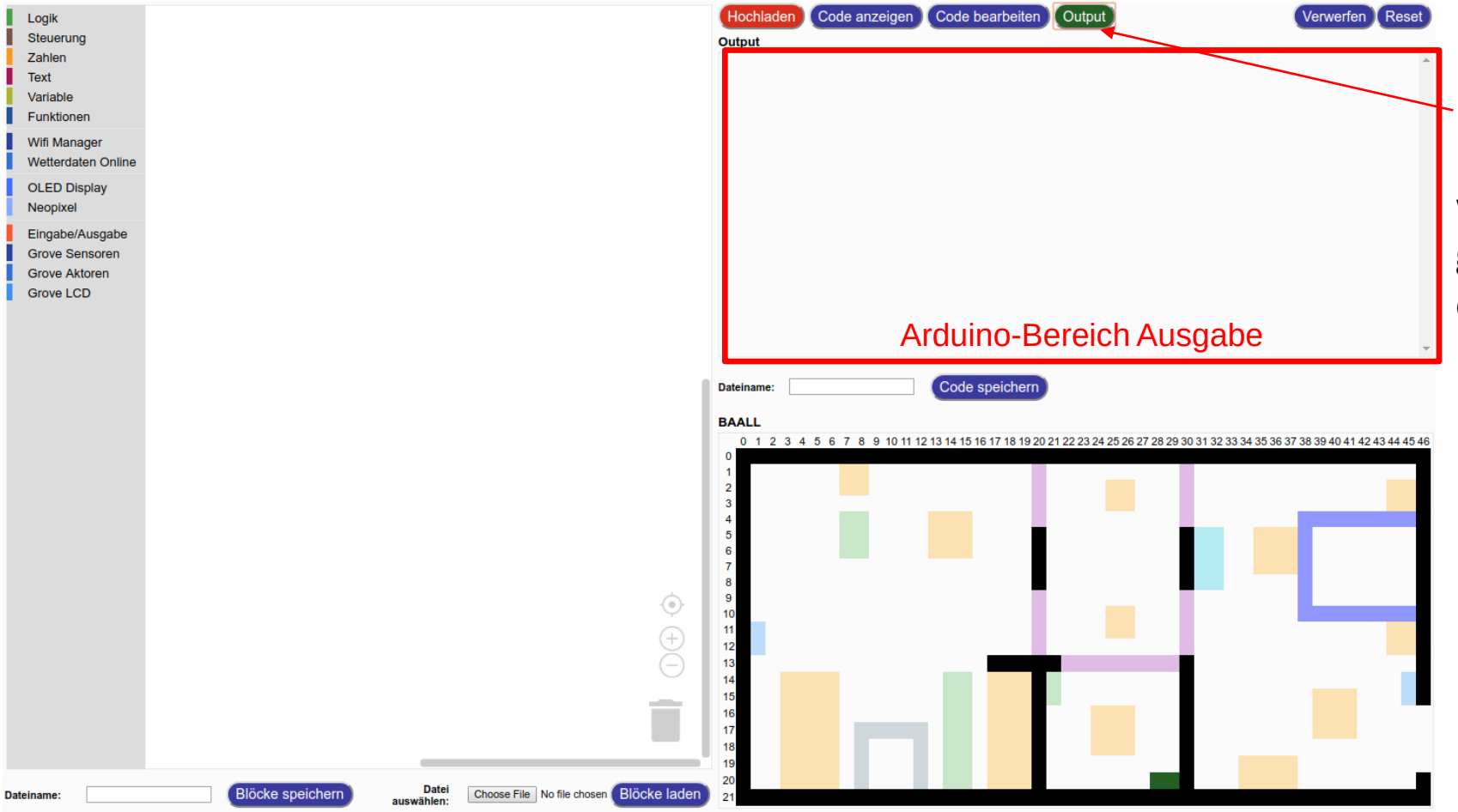
Die Entwicklungsumgebung: BEESM



Wenn das Übertragen erfolgreich war, erscheint „Das Hochladen war erfolgreich“.

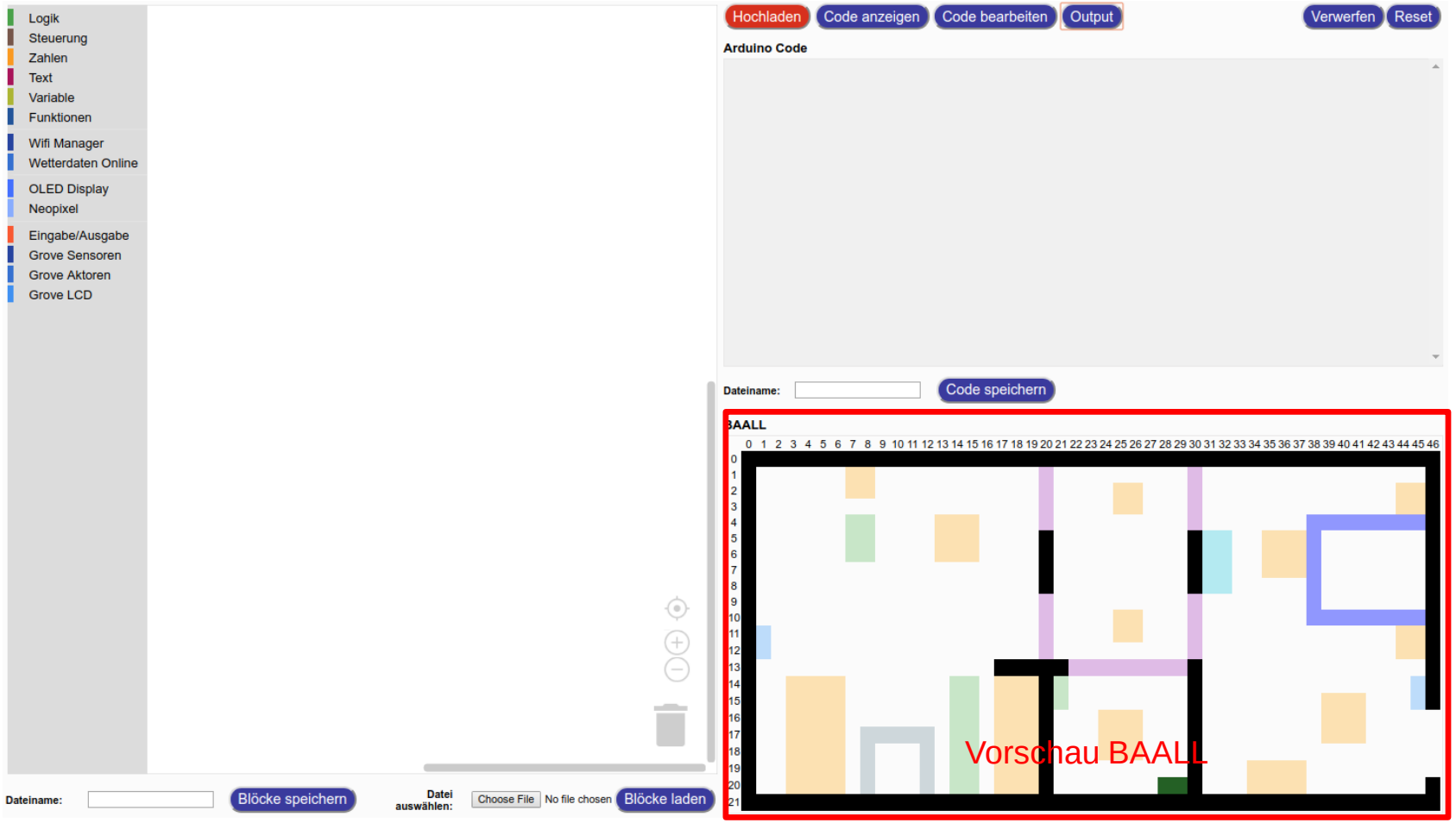
Wenn das Übertragen nicht erfolgreich war, erscheint „Upload war nicht erfolgreich“. Dann muss herausgefunden werden woran es lag.

Die Entwicklungsumgebung: BEESM

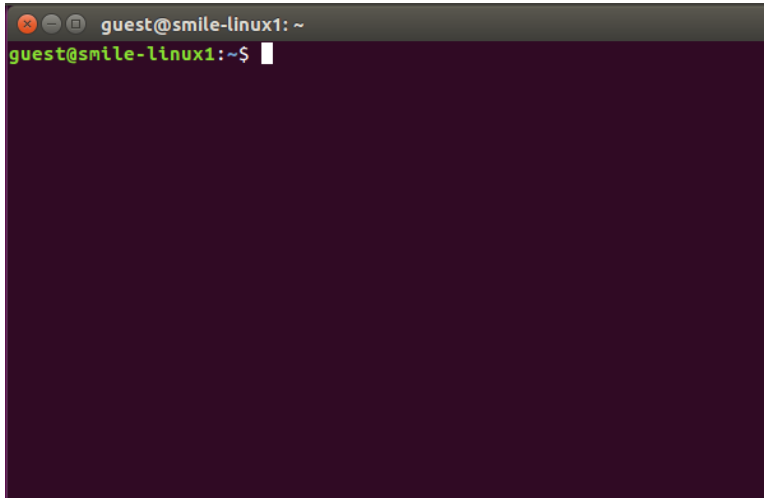


Im Ausgabe Bereich können wir sehen was unser Programm gerade macht. Also ob es uns z. B. einen Fehler ausgibt.

Die Entwicklungsumgebung: BEESM

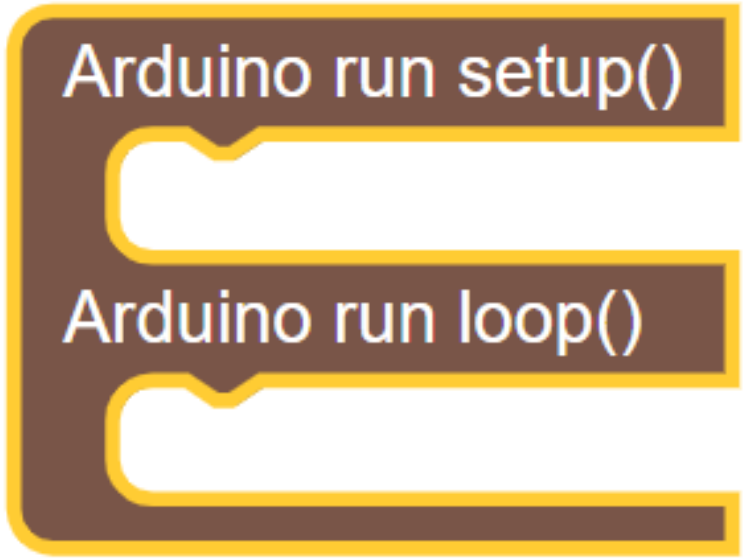


Die Entwicklungsumgebung: BEESM



./start.sh

Arduino Programmstruktur



- Setup:
  - Wird **einmal** zu Beginn des Programms ausgeführt.
- Loop:
  - Wird nach dem Setup als **Endlosschleife** ausgeführt

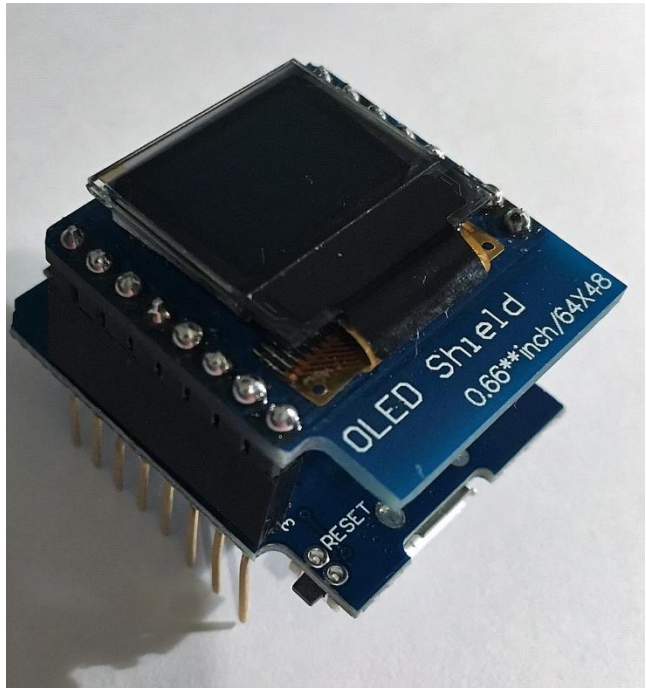
Der Loop

Arduino Code

```
#include <Adafruit_NeoPixel.h>

Adafruit_NeoPixel strip_D3 = Adafruit_NeoPixel(1 , D3 , NEO_GRB + NEO_KHZ800);

void setup()
{
  strip_D3.begin();

  strip_D3.setBrightness(150);

  strip_D3.show();

}


void loop()
{
  strip_D3.setPixelColor(0,200,0,200);
  strip_D3.show();
  delay(1000);
  strip_D3.setPixelColor(0,0,0,0);
  strip_D3.show();
  delay(1000);

}
```

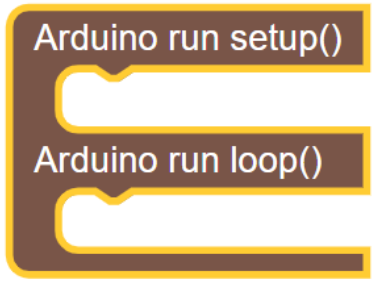
Loop()

Das OLED-Display



Ein erstes Programm

- Schreibe ein Programm,
  das „Hallo Welt“ auf dem Display ausgibt.



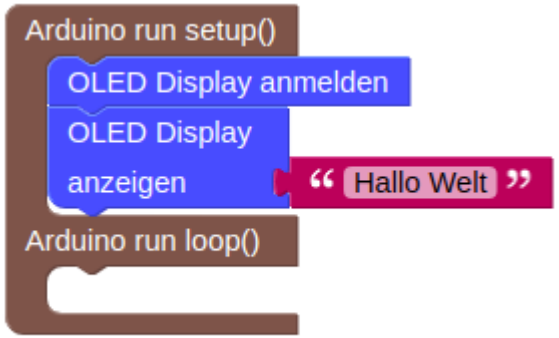
- Beachte dabei folgende Schritte:
  - Ziehe den Block für die Arduino Programmstruktur in den Programmierbereich
  - Ziehe die Anweisungen für die Text-Ausgabe in den Setup Bereich
    - Wir müssen das Display **anmelden** und dann den **Text anzeigen**. Die passenden Befehle findet Ihr unter OLED Display
  - Lade dein Programm in den Controller hoch
  - Der Upload dauert ein paar Sekunden. Warte bis die Bestätigung „Programm erfolgreich in den Controller hochgeladen“ erscheint.

Mein erstes Programm

Das Resultat sollte ungefähr so aussehen:

Mein erstes Programm

Das Resultat sollte ungefähr so aussehen:



Übungen 1: Das Display

- Ändert euer Programm
  - Ändert den Text, den Ihr ausgebt
  - Verändert die Position des Textes mit „Cursor Position“.
  - Ändert die Textgröße

Frage 1: Wieviel Text passt auf das Display bei einer Schriftgröße von 1?

_____ Reihen, _____ Zeichen in einer Zeile

Frage 2: Welche Nummer gebt Ihr im Block „Cursor Position“ für Reihe an, wenn ihr Text in der zweiten, dritten und vierten Zeile ausgeben wollt? (in Schriftgröße 1)

_____

Übungen 2: Das Display

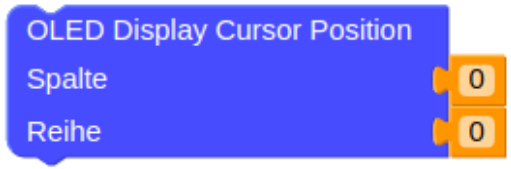
Tipps!

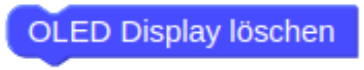
1000 Millisekunden entspricht 1 Sekunde

Den Block „delay“ findet ihr in der Blockpalette unter Steuerung

- Ändert euer Programm
  - Gebe den Text in der Loop aus.
  - Ziehe einen Block für eine Pause 

zusätzlich in die Loop
  - Schreibe ein Programm, das die Worte „Hallo“ und „Welt“ fortlaufend im Wechsel auf dem Display ausgibt. Probiere dazu auch folgende OLED-Display Blöcke aus:

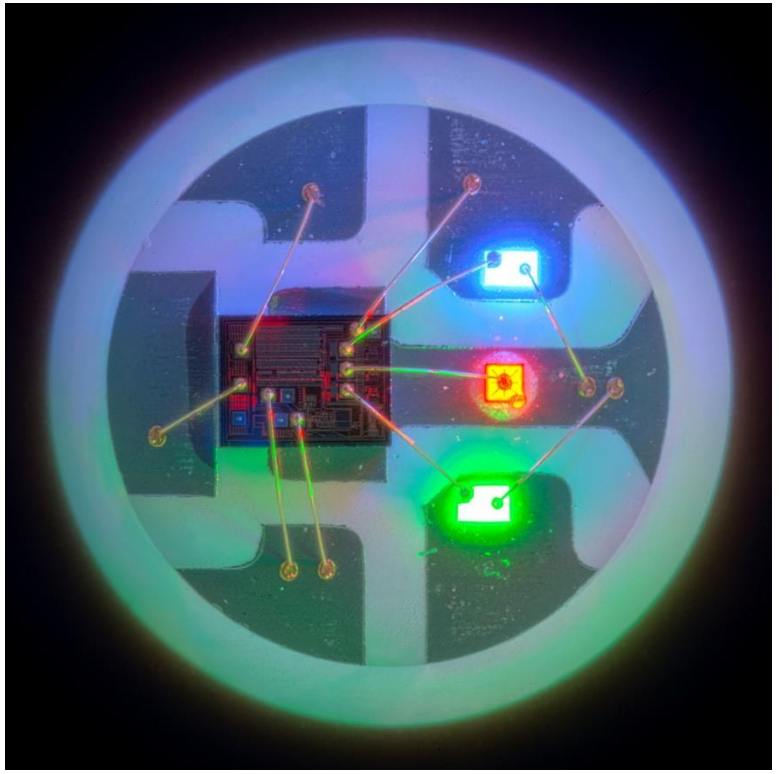




Frage 1: Was passiert, wenn Ihr den Text in der Loop ausgeben lasst? Warum? 

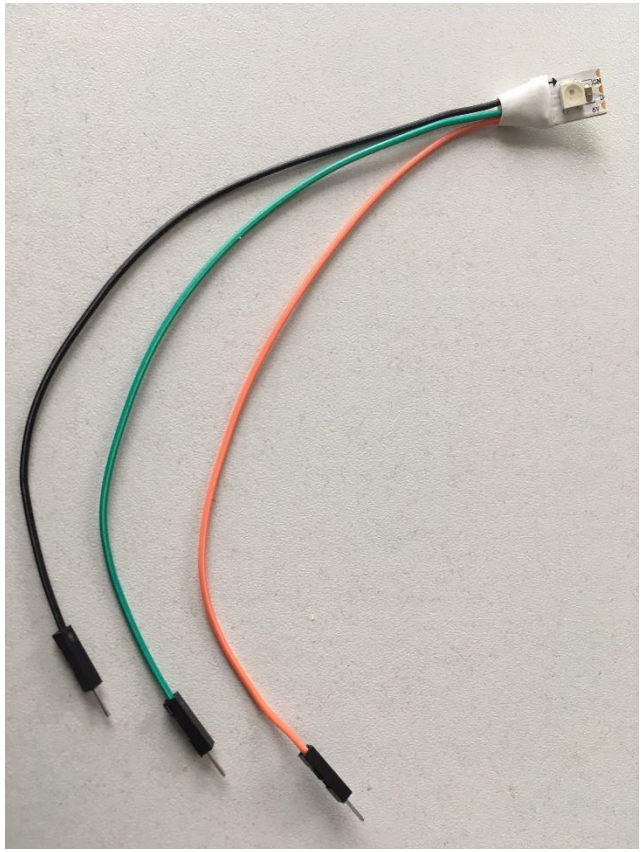
Frage 2: Was bedeutet die Zahl in dem „delay“ Block? 

RGB-LED

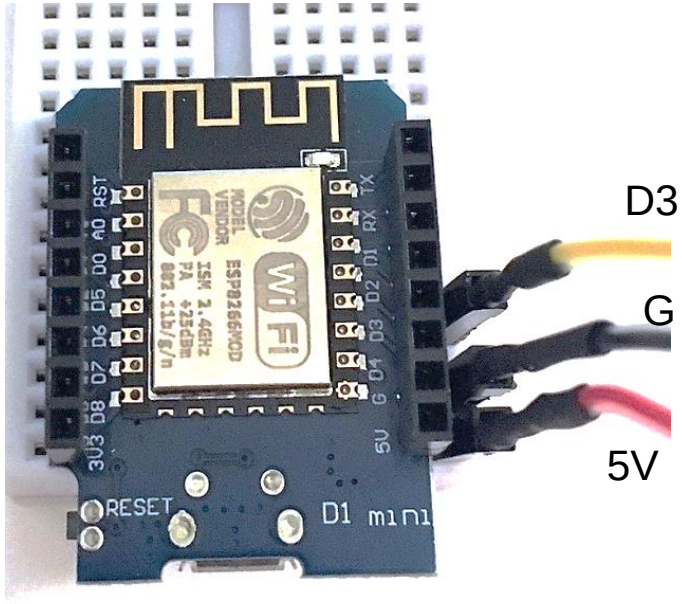


- Neopixel (→ „schlaue“ Pixel)
- Ein Lämpchen hat ein rotes, grünes und blaues Pixel, damit kann man 16,7 Mio. Farben mischen.

RGB-LED-Anschließen



RGB-LED-Anschließen



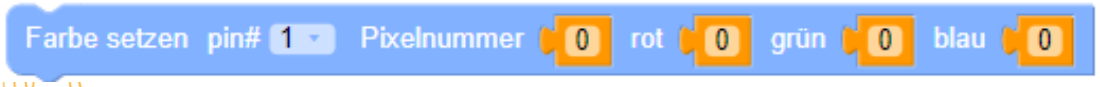
+ Strom (5V)

- Masse, Ground (G, GND)

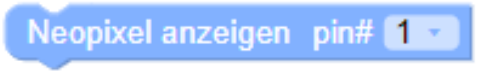
Daten, (D1…Dx)

Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

RGB-LED-Neopixel



Wichtig! Bei allen Blöcken muss die **Pin** angegeben werden, an welchem wir die LED angeschlossen haben



Anmelden: Wir definieren den **Pin**, an welchen wir die Datenleitung (gelbes, grünes oder blaues Kabel) unserer LED-Lämpchen angeschlossen haben und **wieviele LEDs** wir in Reihe geschaltet haben. Dies geschieht 1mal im Setup des Programms.

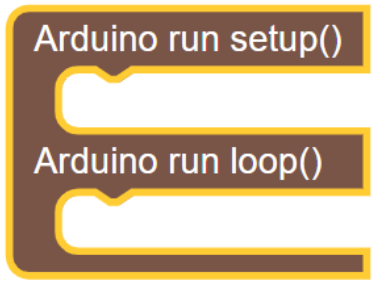
Optional: Wir können die **Helligkeit** der Lämpchen setzen. Ohne Angabe leuchten die Lämpchen mit voller Intensität, das ist gleich 255. Diese Angabe sollte nur einmal am Anfang des Programms im Setup gesetzt werden, wenn wir die Lämpchen insgesamt weniger hell haben wollen.

Farbe setzen: Hier sagen wir dem Programm, in welcher Farbe das LED-Lämpchen leuchten soll. Die Farbe wird aus den Werten für rot, grün und blau gemischt. Die Werte können jeweils zwischen 0 (aus) und 255 (höchste Intensität) liegen. **Pixelnummer** 0 bedeutet, dass wir das erste Lämpchen einschalten.

Die oben gesetzten Farben muss ich nun an die LEDs senden. Sonst passiert nichts. Diese Anweisung brauchen wir nach jeder Änderung der Lämpchen.

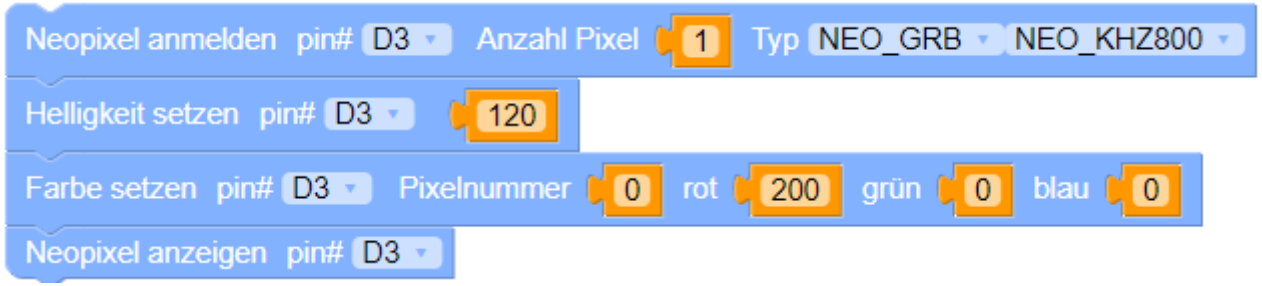
Ein weiteres Programm

- Schreibe ein Programm, dass das Lämpchen in rot leuchten lässt.



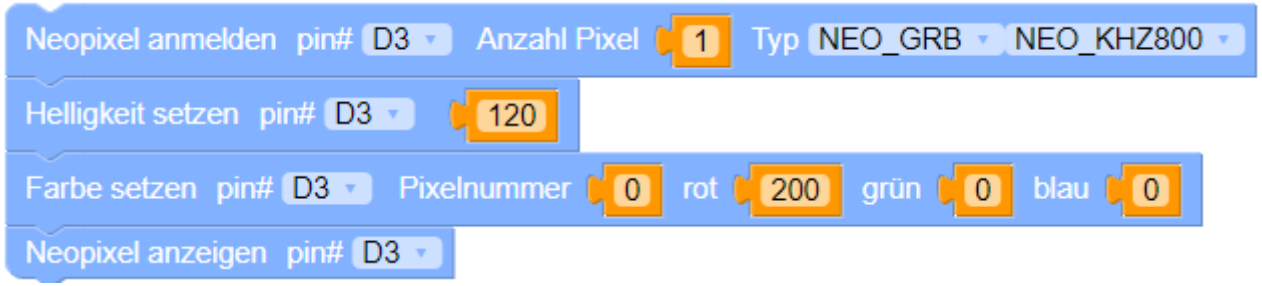
- Beachte dabei folgende Schritte:
  - Schließt das LED Lämpchen an den Controller
  - Ziehe den Block für die Arduino Programmstruktur in den Programmierbereich
  - Wir müssen die LEDs im Setup **anmelden** und dann zum leuchten bringen. Die passenden Befehle findet Ihr unter Neopixel
  - Ladet euer Programm in den Controller hoch

Beispielprogramm



Diese Blöcke 1 mal setzen
am Anfang des Programms

Beispielprogramm



Diese Blöcke 1 mal setzen
am Anfang des Programms

Programmcode in der Schleife:
Wie sollen sich die Lämpchen „verhalten“

Was ist ein Programm?

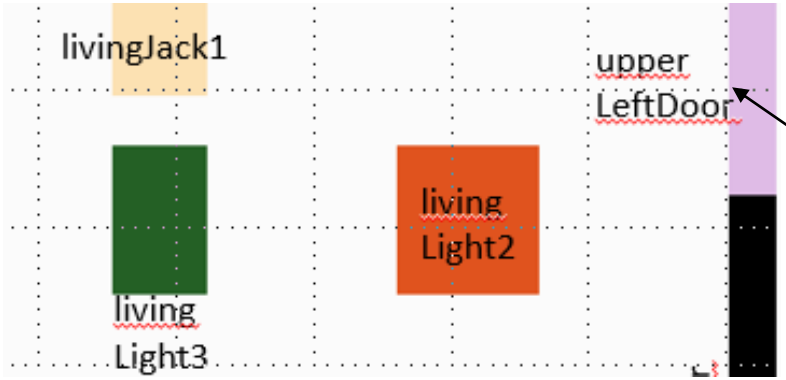
- Habt Ihr da Ideen?

Das BAALL programmieren

Das BAALL programmieren

Auf den folgenden Seiten findet ihr eine Skizze vom BAALL und eine Liste aller Objekte mit Name und Status auf die wir zugreifen können. Zugreifen bedeutet, dass wir den Status einzelner Objekte im BAALL auslesen und ändern können.

Auszug aus der Skizze:



Objekt Name,
z.B. basin

oder

upperLeftDoor

Auszug aus der Liste:

Status Wert, z.B. 108 oder off

```
basin: 108
bathroomLight: off
bathroomToiletHeight: 111
bathroomdoor: on
beaconActive: off
bed1: 108
bed4: 96
bedroomJack1: off
bedroomJack2: off
bedroomLight1: off
bedroomLight2: off
bedroomTVActive: off
bulblamp: 0:0:0
corridorLight: off
```

Aufgabe

Aufgabe: Schaut euch die BAALL Skizze und die Liste an – achtet dabei besonders auf die Werte, die der Status der Objekte annehmen kann. Welche Objekte aus der Liste würdet Ihr den Klassen „Switch“, „Dimmer“, „RGB Farbe“, „Sensoren“ und „TV Programm“ zuordnen? Warum? Notiert/Nennt ein paar Beispiele.

„Switch“: z.B. bathroomdoor, _____

„Dimmer“: z.B. basin, _____

„RGB Farbe“: z.B. bulblamp, _____

„Sensoren“: z.B. LuxOutside, _____

„TV Programm“: _____

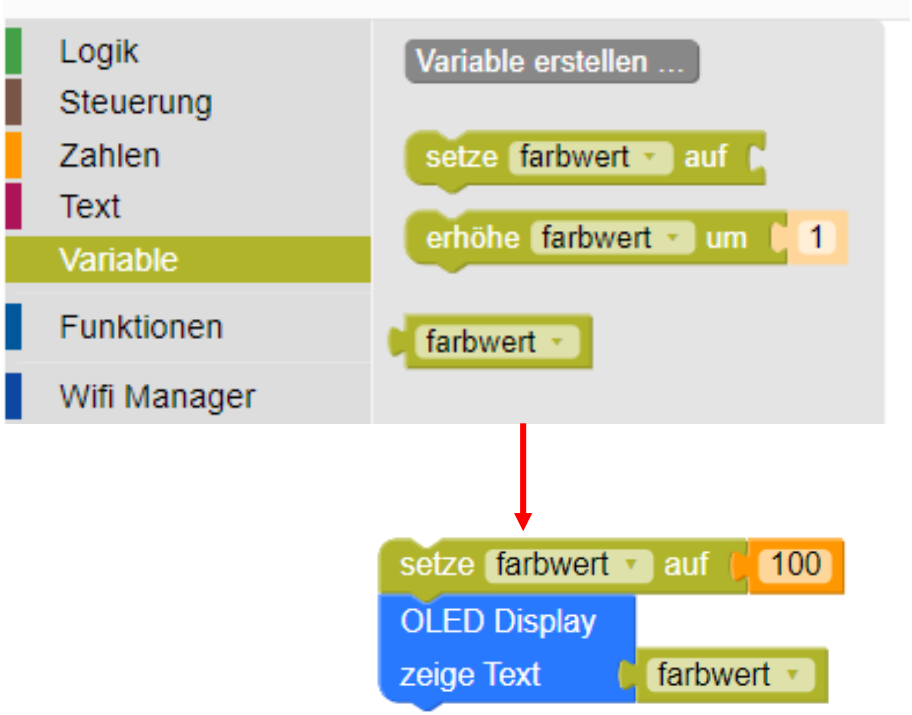
Einführung: Variablen

- Eine Variable ist ein Platz, um Werte zu speichern
- Ähnlich wie ein Karton, in den ich etwas hinein tun kann
- Dieser Karton hat einen eindeutigen Namen, also nur er heißt so
- Bei BEESM können Variablen **NUR** Zahlen enthalten (keinen Text)

Variablen anlegen

- Anlegen eine Variablen im Punkt „Variable“
- Dabei muss der Variable ein Name gegeben werden.
- Danach erscheinen neue Befehle unter „Variable“ für diese Variable

Variablen anlegen



- Variablen können gelesen und beschrieben (gesetzt) werden
- Zum setzen einer Variable nehmt den Block „setze *variablenname* auf *x*“
- Zum Lesen der Variable zieht diese in eine andere Operation rein

Übungen: Variablen

Erstellt eine Variable, speichert einen Wert (Zahl) in der Variablen und lasst den Wert der Variablen auf dem Display ausgeben!

- Was gibt der Code auf dem Display aus?

  _____

Übungen: Variablen

Nun lassen wir den Computer rechnen.

Anstelle einer einzigen Zahl (wie in diesem Beispiel die 3) setzen wir die Variable auf komplexere Terme („mathematische Gebilde“). Entsprechende Blöcke findet ihr unter Zahlen.

Übungen: Variablen

Ändert euer Programm:

- Legt eine Variable an und weist dieser einen Wert zu.
- Benutzt „erstelle Text aus“ um zusätzlich noch einen Text auszugeben
- Gebt den Wert der Variable auf dem Display aus

Tipp !

Den Block, der einen Text aus mehreren Textbausteinen und Zahlen zusammensetzt, findet Ihr unter „Text“.

Übung 2: Variablen

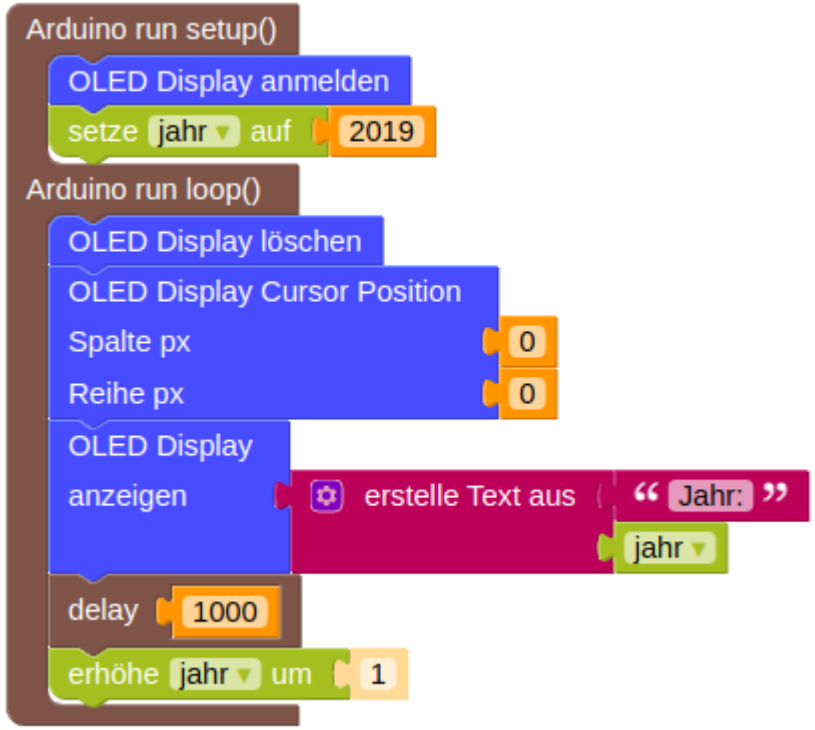
- A) Erweitert euer Programm und programmiert einen Zähler:
  - Wartet nach der Ausgabe der Variable für eine Sekunde
  - Erhöht dann den Wert der Variable um eins 

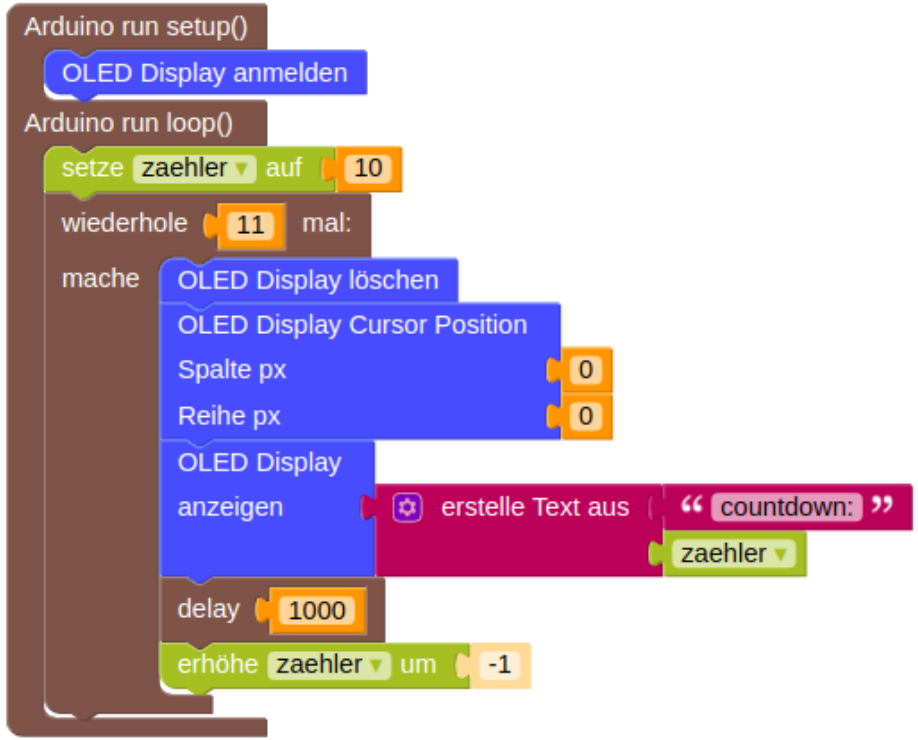
  - Gebt die geänderte Variable aus

B) Programmiert einen Countdown, der von 10 bis 0 runterzählt.

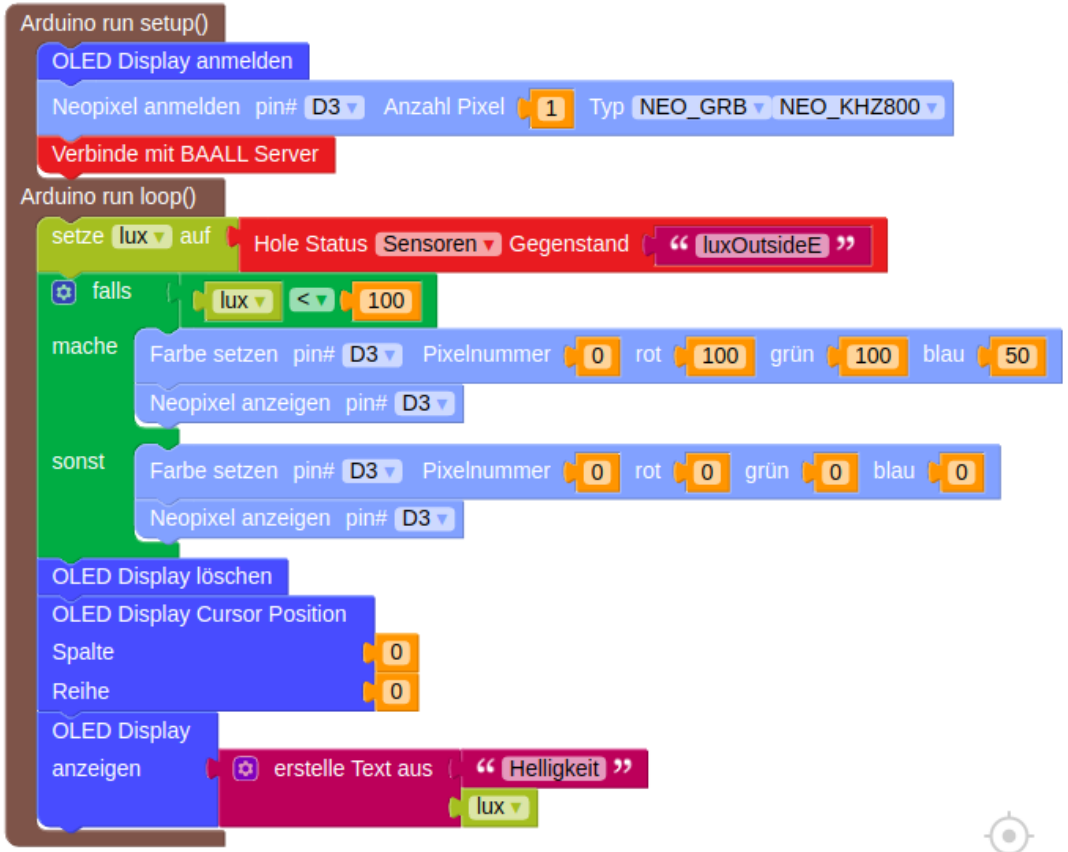
Lösung: Variablen 2

- Ändert euer Programm: Zähler und Countdown





Programm – Intelligente Lichtsteuerung



Was macht dieses Programm?