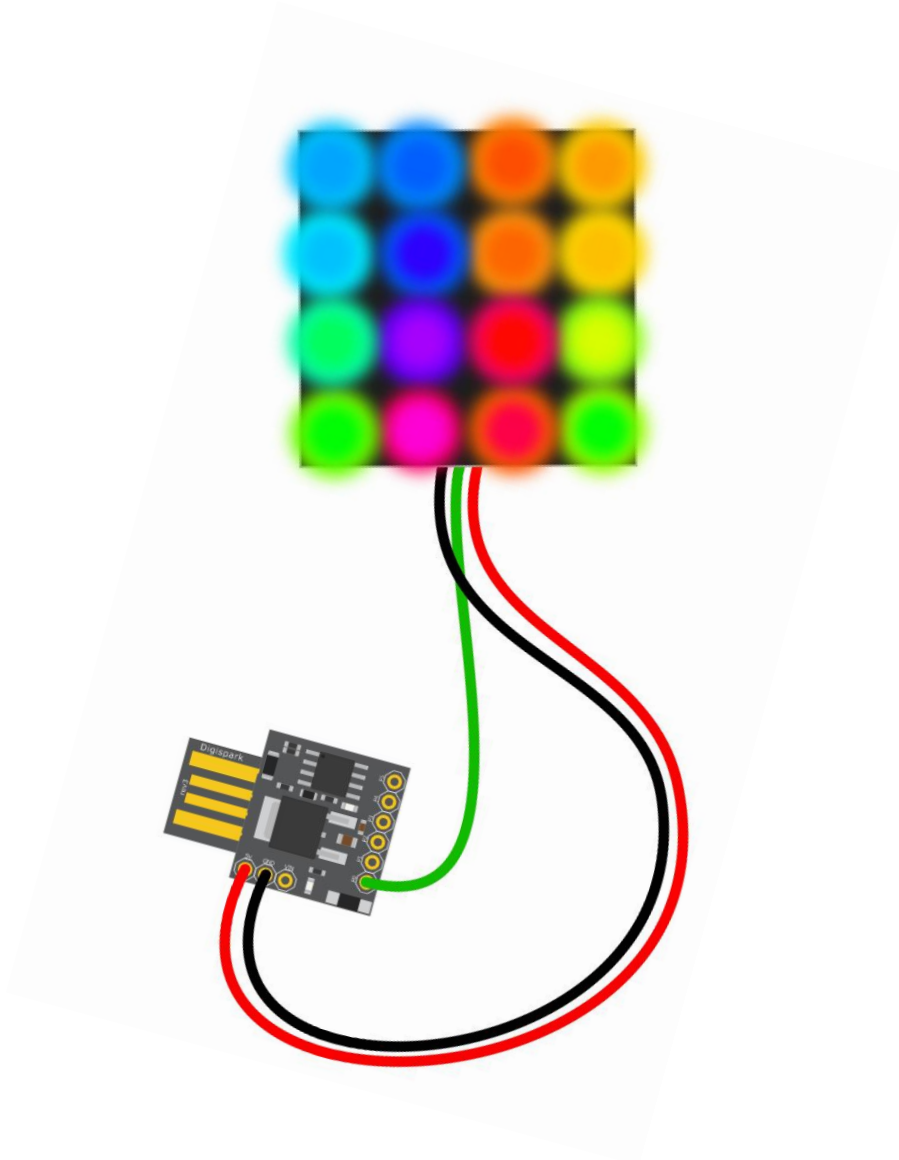


## ATTINY85 MIT PGLU PROGRAMMIEREN



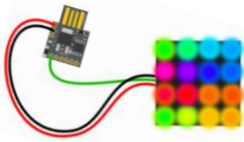
---

Stand 24.1.2021

Dieses Dokument zeigt, wie ein ATtiny85 - das wohl kleinste Arduino-kompatible Mikrocontroller Board - mit dem PGLU-Editor und weiteren Methoden, programmiert wird.

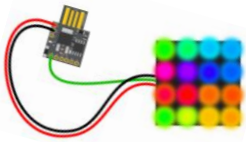
Hinweis: Das Arbeiten mit dem ATtiny85 ist experimentell und es gibt einige Stolpersteine. Es ist zu empfehlen, alle Schritte vorher zu üben und im Unterricht nur mit erprobter Hardware auszuführen! Bei Problemen kann Kapitel 8 helfen.

Hinweis: Die folgenden Anleitungen wurden noch nicht auf einem Mac getestet.



## INHALT

<b>ATtiny85 mit PGLU programmieren</b> .....	1
<b>1. Übersicht</b> .....	3
<b>2. Pin Belegung</b> .....	3
2.1. Pinbelegung der USB-Version von Digispark.....	3
2.2. Pinbelegung der Chip-Version.....	4
<b>3. Sketch mit PGLU-Editor schreiben</b> .....	5
3.1. Sketch anpassen: Pin-Nummern in Blöcken ersetzen .....	5
3.2. Sketch anpassen: Neopixel Daten-Pin ändern .....	6
<b>4. Sketch auf USB-Version des Attiny85 übertragen</b> .....	7
4.1. PC/Mac vorbereiten .....	7
4.2. Sketch aus PGLU-Editor an Board übertragen .....	8
<b>5. Sketch auf Chip-Version des ATtiny85 hochladen</b> .....	9
<b>6. Beispielprojekt Sparkling Stars</b> .....	9
6.1. Sketch.....	9
6.2. Anschlusschema Neopixel .....	10
<b>7. Fragen und Antworten zum ATtiny85</b> .....	11
<b>8. Problemlösungen</b> .....	11
<b>9. Links</b> .....	12



## 1. ÜBERSICHT

Der ATtiny85 kann, wie jeder Arduino Mikrocontroller, mit der Sprache C++ programmiert werden. Dies geschieht über den PGLU-Editor, einen anderen grafischen Editor oder direkt mit der Arduino IDE. Obwohl dieser Mikrocontroller winzig klein ist, besitzt er 6 Pins, sowie einen USB-Stecker. Damit kann er also direkt in eine Powerbank oder einen anderen USB-Anschluss gesteckt werden, um mit Strom versorgt zu werden!

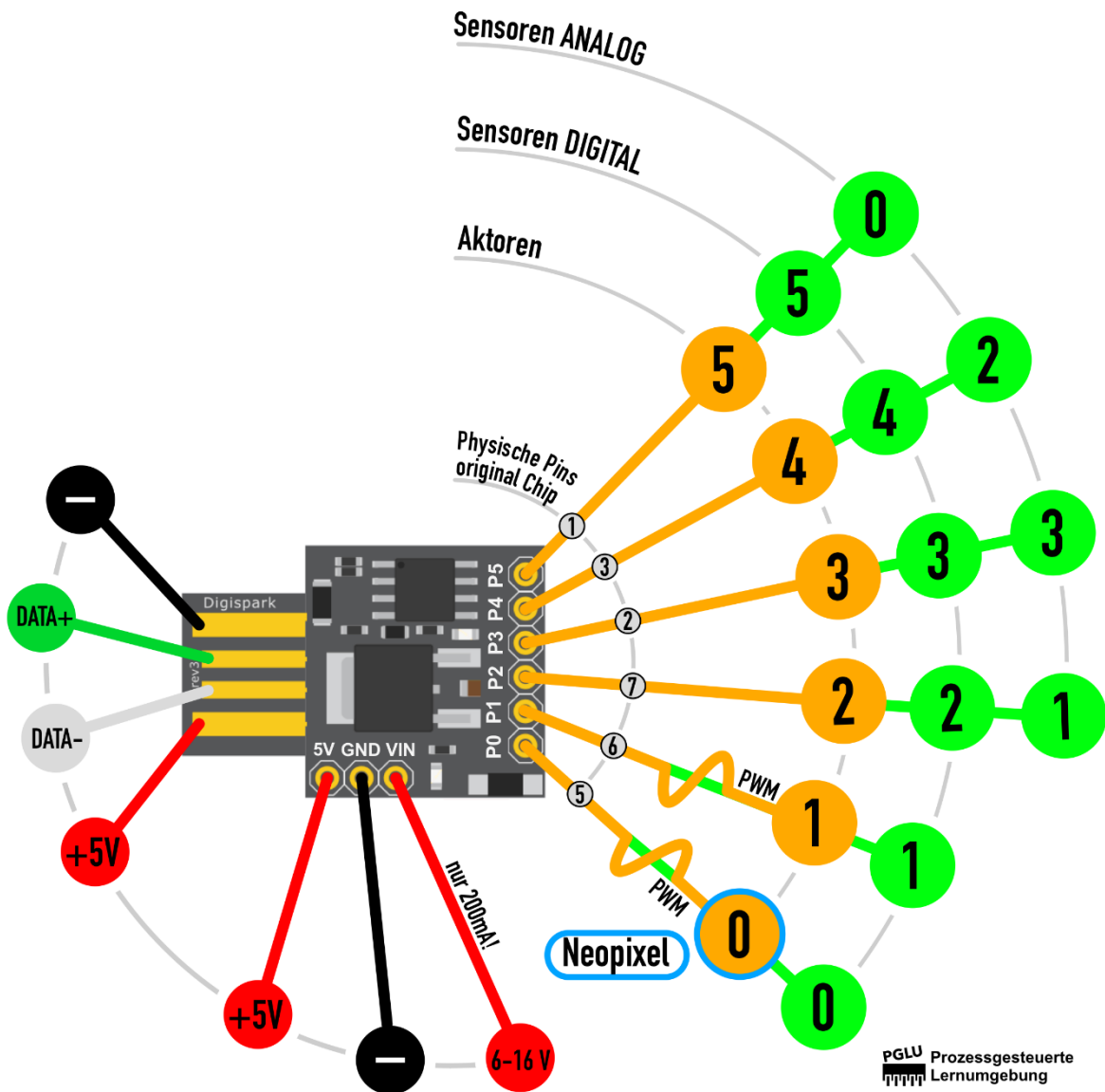
## 2. PIN BELEGUNG

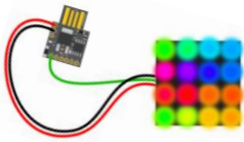
### 2.1. PINBELEGUNG DER USB-VERSION VON DIGISPARK

Wie bei jedem Arduino, können die Pins als Ein- oder als Ausgänge programmiert werden, wobei nur die Pins P0 und P1 ein analoges (PWM-) Signal ausgeben.

Merke: Je nachdem, ob du die Sensoren analog oder digital ausliest, haben die Pins andere Nummern!

Merke: Der Neopixel-Pin 9 aus dem PGLU-Editor muss im C++ Code nachträglich auf Pin 0 geändert werden (siehe Kapitel 3.2)!





## 2.2. PINBELEGUNG DER CHIP-VERSION

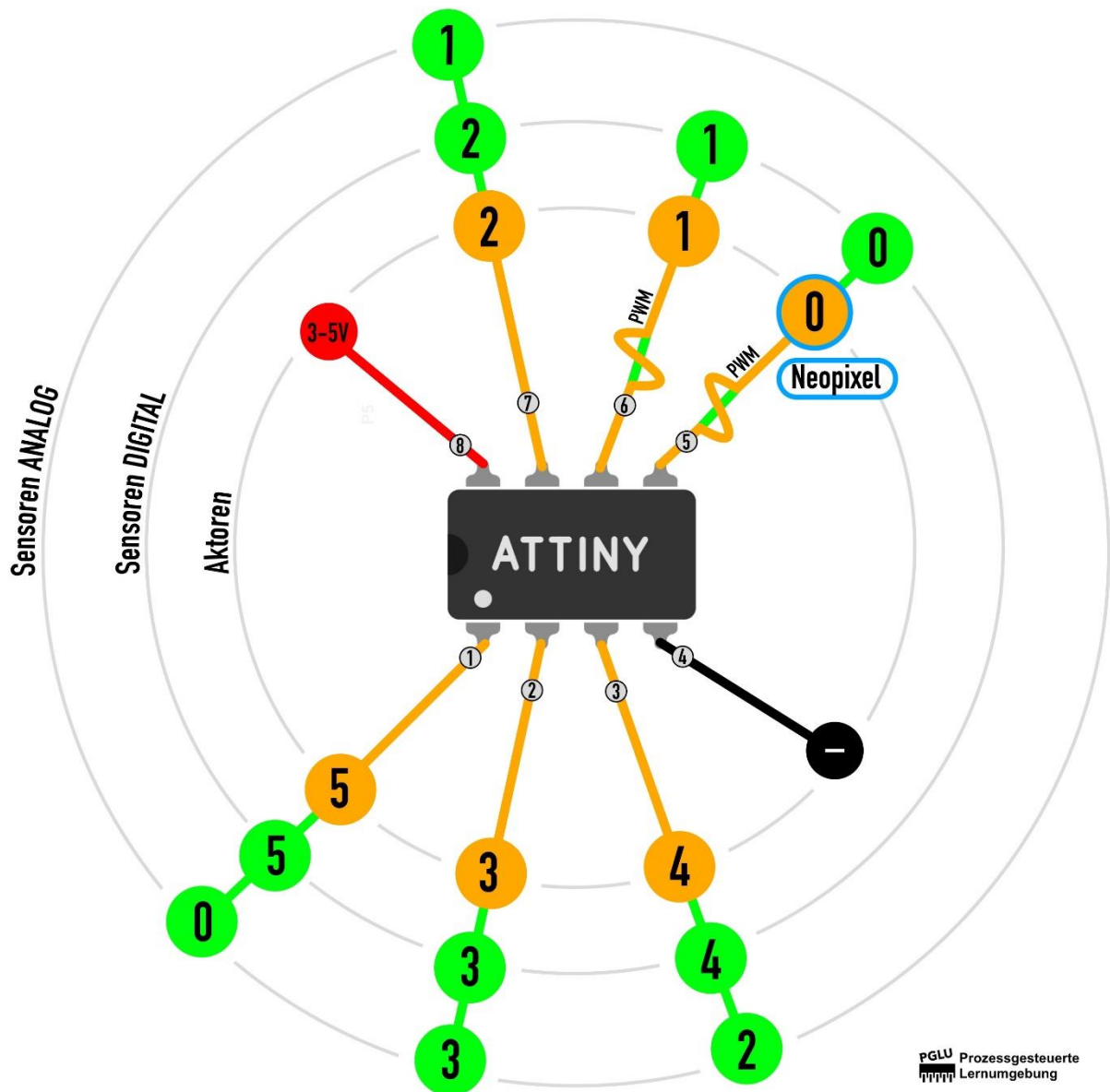
Die Chip-Version des ATtiny85 unterscheidet sich nur durch den fehlenden USB-Anschluss von der oben beschriebenen USB-Variante. Demgegenüber kann dieser Mikrocontroller aufgrund seiner Winzigkeit - die Länge beträgt nur 9mm! - wohl überall eingebaut werden.

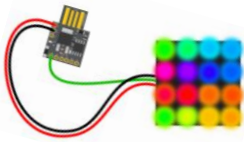
Tipp: Experimentiere mit der virtuellen Chip-Version unter [tinkercad.com/learn/circuits](https://tinkercad.com/learn/circuits)

Hinweis: Da die Chip-Version keinen USB-Anschluss aufweist, kann er nicht an einen PC oder Mac angeschlossen werden. Die Programmierung erfolgt über einen Arduino Mikrocontroller. Eine Anleitung dazu findest du mit dem Keyword: [attiny85 mit arduino nano programmieren](#).

Folgende Materialien werden benötigt:

- > 1 Arduino Nano 3.0
- > 1 ATtiny85 Chip
- > 1 Breadboard
- > 1 ELKO 10 uF





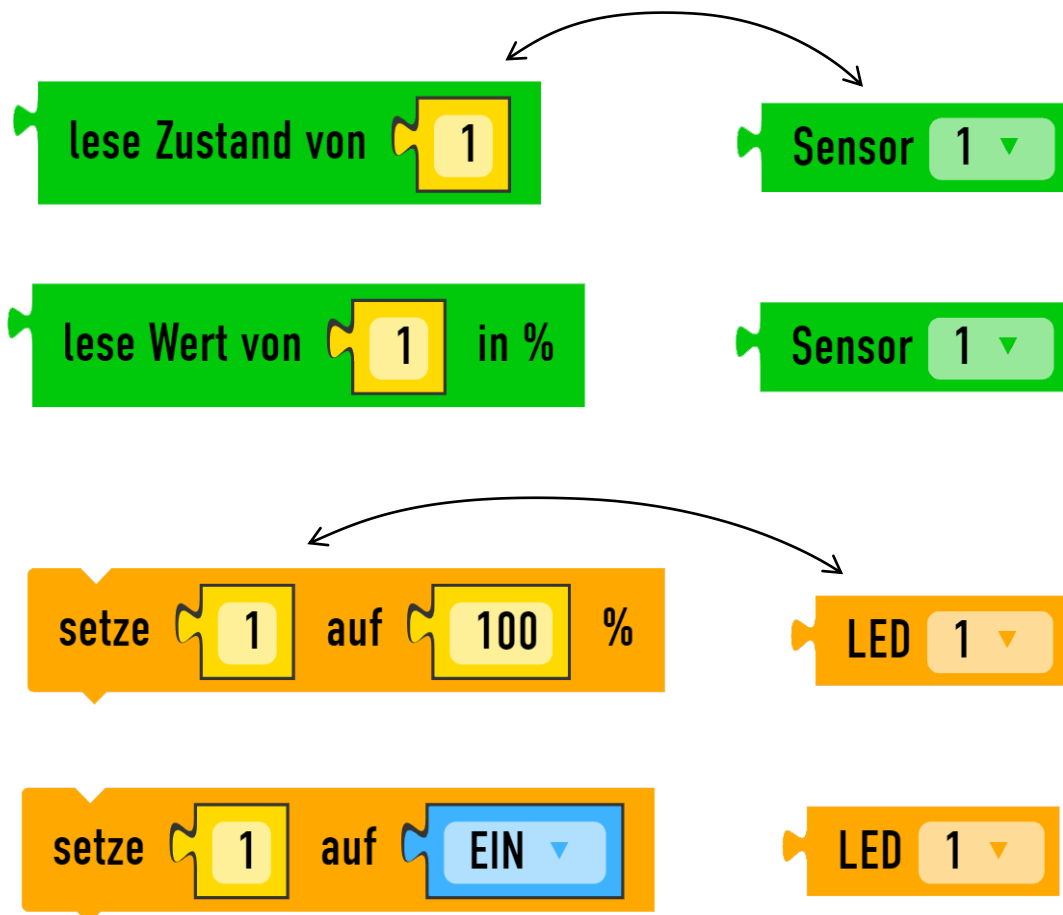
### 3. SKETCH MIT PGLU-EDITOR SCHREIBEN

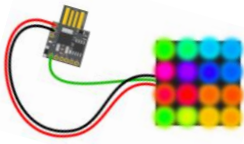
Der ATtiny85 wird mit der gleichen Sprache programmiert, wie jedes andere Arduino: mit C++. Damit können also PGLU Sketches problemlos auch auf dem ATtiny85 laufen gelassen werden.

Damit dies möglich ist, müssen jedoch am Sketch und am C++ Code kleine Anpassungen gemacht werden.

#### 3.1. SKETCH ANPASSEN: PIN-NUMMERN IN BLÖCKEN ERSETZEN

- > Die Pin-Nummern der PGLU-Mikrocontroller (S1, S2, etc.) müssen in die ATtiny85-Pin-Bezeichnungen geändert werden (Kap. 2). Dazu werden bei den Sensor-/Aktor-Bezeichnungen die PGLU-Bezeichnungen entfernt und durch Nummern aus dem Menu «Mathe» ersetzt. Diese Nummern müssen den Bezeichnungen aus Kapitel 2 entsprechen.
- > Beachte, dass die Pin-Nummern ändern, je nachdem, ob du analoge oder digitale Sensoren (lese Zustand.. oder lese Wert..) verwendest (Kap. 2)!





### 3.2. SKETCH ANPASSEN: NEOPIXEL DATEN-PIN ÄNDERN

- > Im Sketch musst du noch den Daten-Pin für die Neopixel anpassen
- > Dieser ist beim Attiny «0» anstatt «LED 1»

Sketch: mein schönes Neopixel Lauflicht

definiere LED-Strip Anzahl Pixel 16 Helligkeit total 50 % Ausgang 0

Hauptloop: 100'000 mal pro Sekunde Blinkcode: kurz 1 lang 1

zähle Pixel von 1 bis 16 alle 1

setze LED-Strip ab Pixel 1

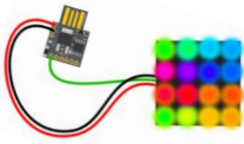
Anzahl 1

auf Farbe in ° 205

Sättigung in % 100

Helligkeit in % 100

pausiere 100 ms

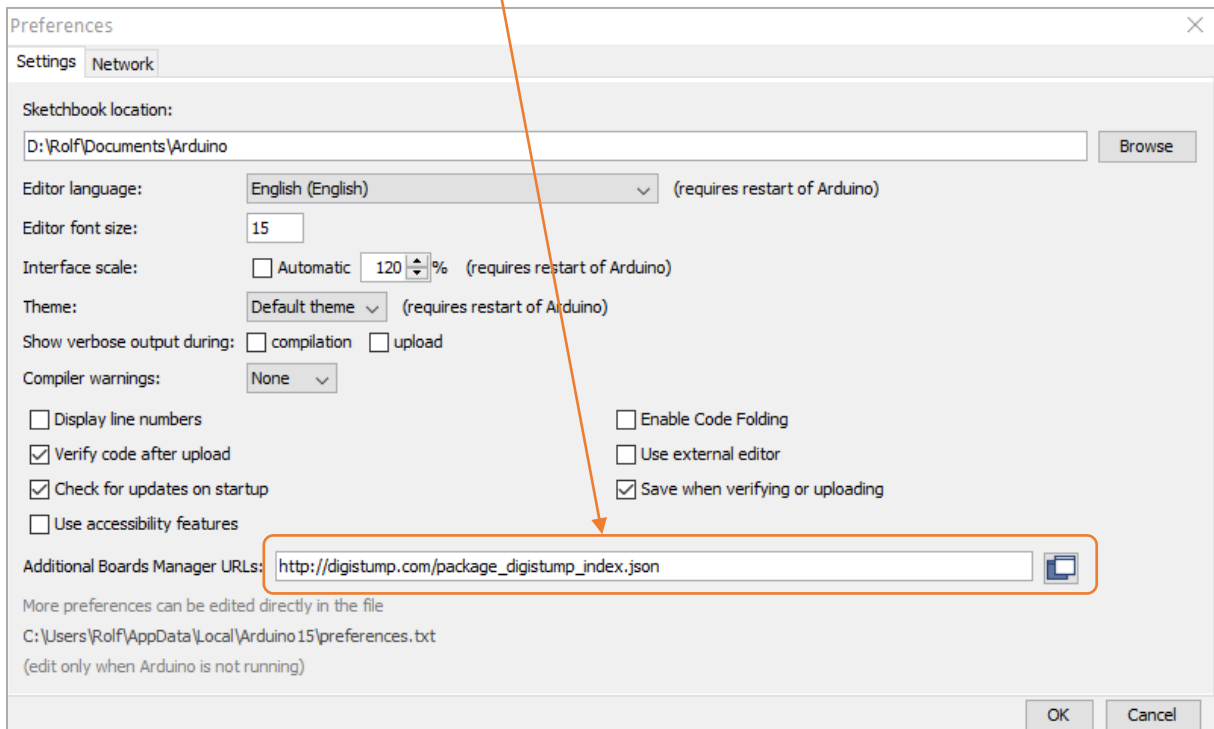


## 4. SKETCH AUF USB-VERSION DES ATTINY85 ÜBERTRAGEN

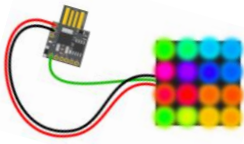
### 4.1. PC/MAC VORBEREITEN

- Treiber für ATtiny85 von Digispark installieren (Windows10): [pglu.ch/micronucleus-win](http://pglu.ch/micronucleus-win) (Install.exe)
- Arduino IDE herunterladen und installieren. [Arduino.cc](http://Arduino.cc) > Software > Downloads
- Arduino IDE öffnen. Dann in: File > Preferences folgenden Eintrag machen

[http://digistump.com/package\\_digistump\\_index.json](http://digistump.com/package_digistump_index.json)



- Board installieren: Tools > Board > Boards Manager...
- Eingabe in Suchfeld: **Digistump AVR Boards**
- Klicke **Install**
- Einstellen: Tools > Board > Digistump AVR Boards > **Digispark (Default - 16.5mhz)**
- Einstellen: Tools > Programmer > **Micronucleus**



## 4.2. SKETCH AUS PGLU-EDITOR AN BOARD ÜBERTRAGEN

- a. C++ Code mit Copy-Paste aus PGLU Editor in Arduino IDE übertragen (siehe Kap. 3)
- b. Wenn du mit Neopixeln arbeitest: Data\_Pin von 9 auf 0 ändern (siehe Kap. 3.2)
- c. ATtiny85 **NICHT** in USB-Anschluss an PC oder MAC einstecken. **Ausgesteckt lassen!**
- d. In der Arduino IDE **Upload (ctrl-U)** klicken und Anweisung im orangen Text befolgen.
  - d.1. Sobald Aufforderung erscheint: ATtiny85 innert 5 Sekunden einstecken
  - d.2. Warten bis Bestätigung «Micronucleus done. Thank you!» erscheint
  - d.3. Fertig
  - d.4. Wenn es nicht Klappt: gehe zu Kapitel 8

```
ATtiny85-Neopixel | Arduino 1.8.13
File Edit Sketch Tools Help
ATtiny85-Neopixel
#include <FastLED.h>

#define NUM_LEDS (constrain(int(16),1,288))
#define DATA_PIN 0
#define BRIGHTNESS constrain(map(50, 0, 100, 0, 255), 0, 255)
CRGB leds[NUM_LEDS];

void setup() {

    FastLED.addLeds<NEOPIXEL, DATA_PIN>(leds, NUM_LEDS);
    FastLED.setBrightness(BRIGHTNESS);
}

Uploading...
D:\Rolf\Documents\Arduino\libraries\FastLED\src\fastspi.h:135:23: note: #prag
# pragma message "No hardware SPI pins defined. All SPI access will de
^
Sketch uses 5842 bytes (97%) of program storage space. Maximum is 6012 bytes
Global variables use 139 bytes of dynamic memory.
Running Digispark Uploader...
Plug in device now... (will timeout in 60 seconds)

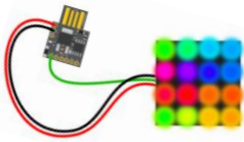
19 Digispark (Default - 16.5mhz) on COM5
```



```
Done uploading
> Starting to upload
writing: 70% complete
writing: 75% complete
writing: 80% complete
> Starting the user app ...
running: 100% complete
>> Micronucleus done. Thank you!

19 Digispark (Default - 16.5mhz) on COM5
```





## 5. SKETCH AUF CHIP-VERSION DES ATTINY85 HOCHLADEN

Das Hochladen eines Sketches auf die Chip-Version wird hier nicht näher beschrieben. Eine Anleitung dazu findest du mit dem Keyword: [attiny85 mit arduino nano programmieren](#).

Folgende Materialien werden benötigt:

- > 1 Arduino Nano 3.0
- > 1 ATTiny85
- > 1 Breadboard
- > 1 ELKO 10 uF

## 6. BEISPIELPROJEKT SPARKLING STARS

### 6.1. SKETCH

1. Sketch programmieren:

definiere LED-Strip Anzahl Pixel  Helligkeit total  %

Hauptloop: 100'000 mal pro Sekunde

setze **Zwischenspeicher** ab Pixel ganzzahliger Zufallswert zwischen  bis

Anzahl

auf Farbe in ° ganzzahliger Zufallswert zwischen  bis

Sättigung in %

Helligkeit in %

pausiere  ms

zähle Pixel von  bis  alle

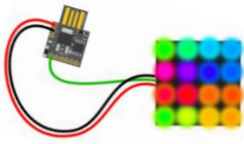
dimme Pixel  im Zwischenspeicher um %

sende Pixel aus Zwischenspeicher an LED-Strip

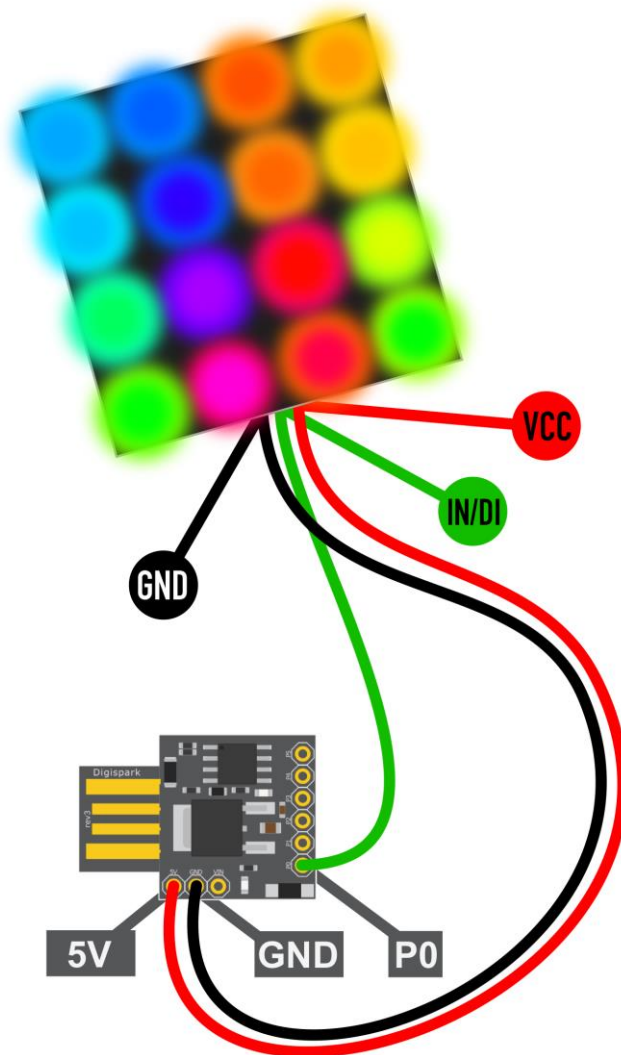
**Beachten!**

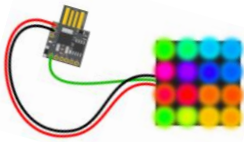
2. Sketch als C++ Code in Arduino IDE übertragen und Data\_Pin von 9 auf 0 ändern

3. Siehe Video unter: [vimeo.com/504084823](https://vimeo.com/504084823)



## 6.2. ANSCHLUSSSCHEMA NEOPIXEL



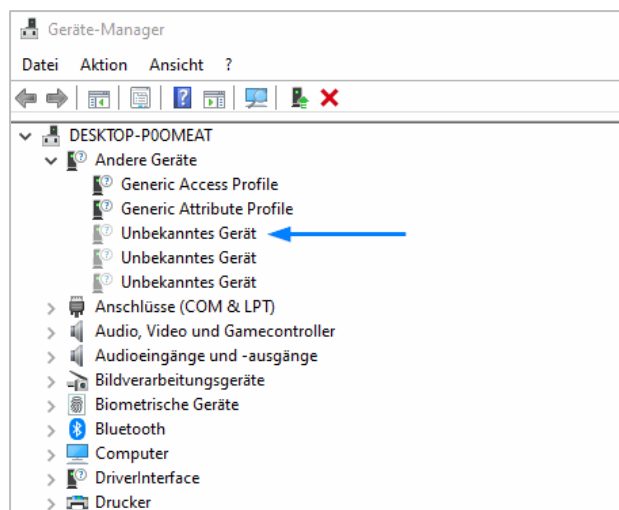


## 7. FRAGEN UND ANTWORTEN ZUM ATTINY85

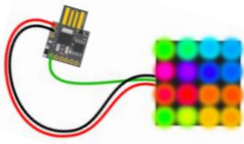
- > Kann jeder Pin wahlweise als Eingang oder als Ausgang verwendet werden?
  - > Ja, das geht
- > Können an den ATtiny85 Motoren angeschlossen werden?
  - > Nein. Die Ausgangs-Pins können mit maximal 40mA belastet werden. Gleich wie beim Nano.
- > Kann der ATtiny85 an die PGLU Teacher's Box angeschlossen werden?
  - > Nein, das geht nicht. Der ATtiny85 kann in der USB-Version aber an jeden PC oder Mac angeschlossen werden. Die Chip-Version wird mit einem Arduino Nano und einem Breadboard programmiert.

## 8. PROBLEMLÖSUNGEN

- > Windows Rechner erkennt Digispark ATtiny85 nicht (Upload nicht möglich)
  - > Treiber installieren unter: [pglu.ch/micronucleus-win](http://pglu.ch/micronucleus-win) (Install.exe)
- > Windows Rechner erkennt Digispark ATtiny85 nicht (Upload nicht möglich)
  - > Der Digispark ist nur für 5 Sekunden nach dem Einstecken für Uploads verfügbar. Erfolgt danach eine «Enumeration», so wird in der Regel ein unbekanntes Gerät angezeigt.
  - > Stecke den Digispark aus. Gehe in den Geräte-Manager und lasse dir «Ansicht > Ausgeblendete» Geräte anzeigen. Deinstalliere alle unbekanntes Geräte unter «Andere Geräte» mit einem Rechtsklick auf das Gerät! Starte den PC neu. Stecke den Digispark nur dann ein, wenn du von der Arduino IDE dazu aufgefordert wirst, wenn du versuchst, einen neuen Sketch hochzuladen.
  - > Jedes Mal, wenn du ein unbekanntes Gerät siehst, musst du diese Schritte wiederholen.



- > Windows Rechner erkennt Digispark ATtiny85 nicht (Upload nicht möglich)
  - > Es kann sein, dass der USB-Stecker des Digispark keinen guten Kontakt zur USB-Buchse deines PCs schließt. Eventuell muss der Digispark während dem Upload-Vorgang aktiv etwas schräg in der Buchse gehalten werden.  
Du erkennst, ob der Kontakt richtig geschlossen ist, wenn dein PC ein akustisches Signal beim Ein- und Ausstecken von sich gibt



## 9. LINKS

- > ATtiny und Arduino IDE
  - > <https://playground.boxtec.ch/doku.php/arduino/attiny>
- > Erste Schritte mit ATtiny85 und 84
  - > <https://www.heise.de/ct/artikel/Erste-Schritte-mit-den-Mikrocontrollern-ATtiny84-und-85-4399393.html>
- > ATtiny mit Simulator für elektrische Schaltung und Programmierung
  - > <https://www.tinkercad.com/> > Circuits