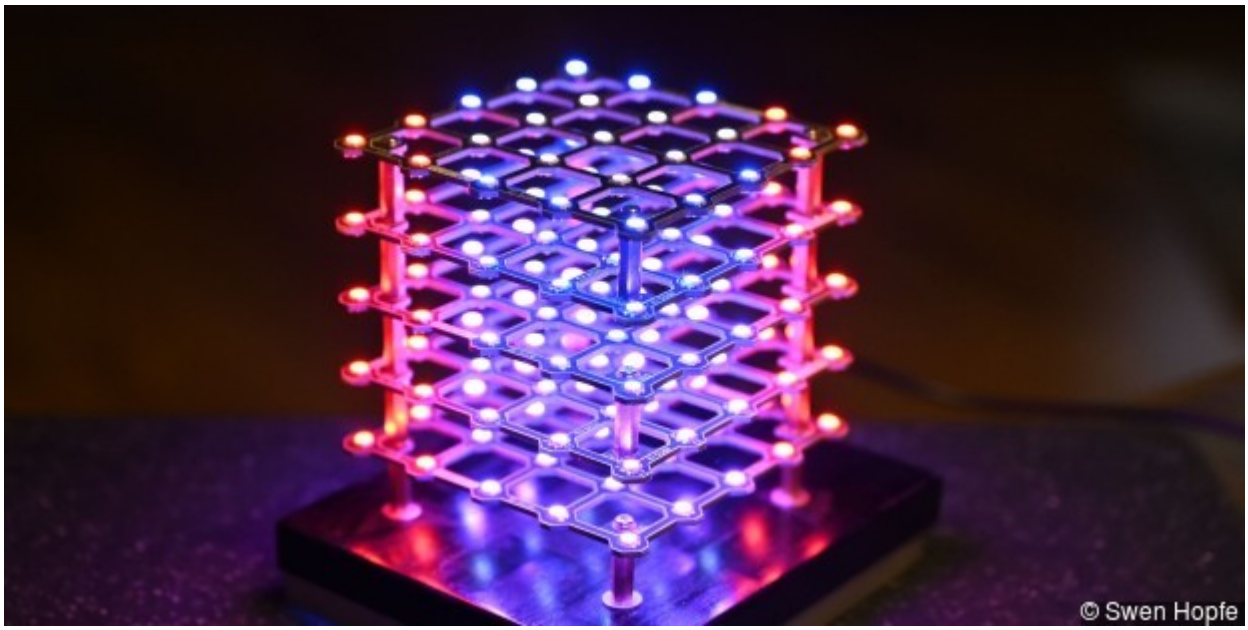
Raspberry Pi Zero steuert Neopixel LED-Cube

08.07.2019 | 10:02 Uhr | Swen Hopfe

Bauen Sie sich einen LED-Cube und steuern Sie ihn mit einem Raspberry Pi Zero an. So geht's.



Vergrößern Cube mit Neopixel-Elementen
© Swen Hopfe

Nachdem ich schon einige Modelle von LED-Würfeln im Eigenbau konstruiert hatte, bin ich auf den Cube:bit aufmerksam geworden. Sein Konzept unterscheidet sich zu anderen LED-Cubes insofern, dass man hier auf den Einsatz von Neopixel-Elementen setzt.

So heißen die mittlerweile bekannten RGB-LEDs, die sich über eine Datenleitung einzeln ansprechen und in Farbe und Helligkeit verändern lassen. Es gibt strang- und ringförmige Module und ganze Flächen. Seit einiger Zeit sind auch sogenannte Slices erhältlich, die aber weniger eine Scheibe als vielmehr ein einzelnes Gitter zum Aufbau einer Würfelebene sind. Von 4tronix entwickelt, werden diese im Handel beispielsweise über Pimoroni angeboten, und es gibt sie mit 3er-, 4er- und 5er-Kantenlängen.

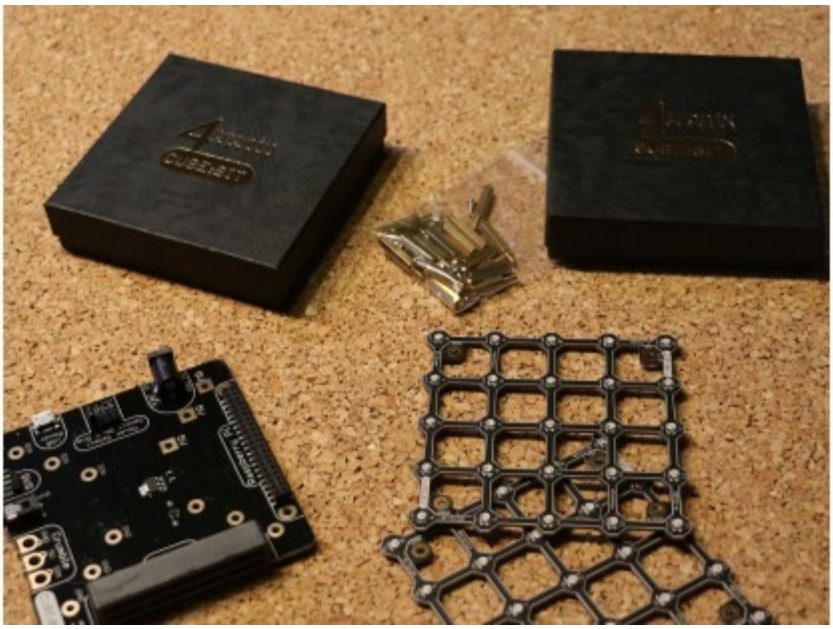
Vergrößern Die Slices, die verwendet werden
© Swen Hopfe

Die Eigenschaften der fertigen Module bringen einige Vorteile mit sich. So entfallen zusätzliche Schieberegister, Controller und LED-Treiber wie die Verdrahtung umfangreicher Zusatzelektronik. Man muss sich nicht um das Multiplexen der einzelnen Ebenen kümmern, und es treten weniger Helligkeitsunterschiede bei unterschiedlicher Anzahl von leuchtenden LEDs auf.

Für den Maker war es bisher immer recht schwierig, ein gleichmäßiges Gitter zu konstruieren, damit der Eigenbau auch optisch was hermacht. Schließlich müssen bei einem 5x5x5-Würfel schon 125 LEDs geprüft, gebogen und mit weiteren Drähten möglichst akkurat verlötet werden. Bei noch größeren Würfeln wird das schnell zu einer Mammut-Aufgabe. Das wird einem nun alles abgenommen.

Nachteile hat diese spezielle Lösung natürlich auch. Sofort offensichtlich ist die stärkere Verstrebung zwischen den LEDs, welche beim Betrachten auch Lichtpunkte ausblendet. Man hat auf dem Leiterplattenmaterial jeweils oben und unten eine LED angeordnet, die beide gleichmäßig angesteuert werden, was das Ganze etwas abmildert, aber ein schlanker Drahtaufbau ist da transparenter.

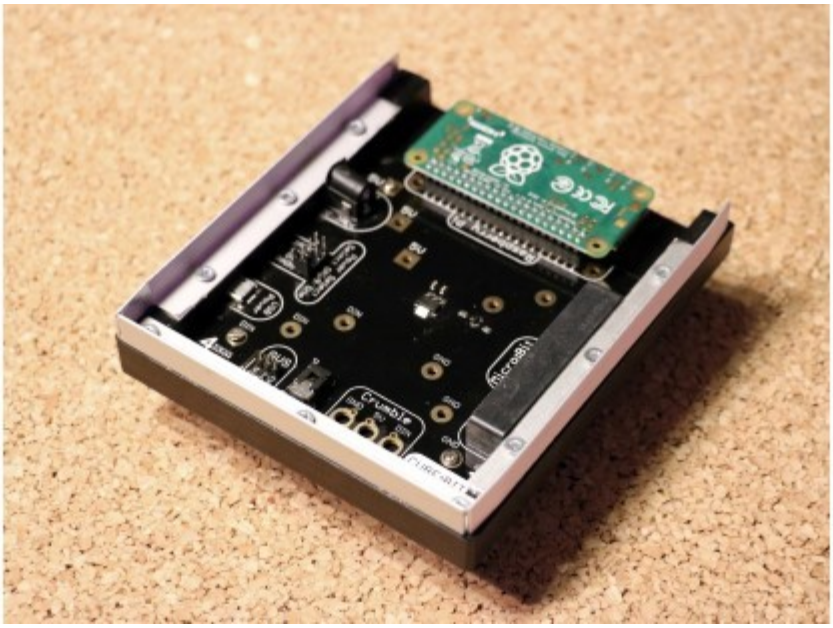
Und dann ist da der im Moment noch recht hohe Preis für einen Bausatz und für nachzukaufende Slices. Durch die vorgegebene Adressierung der Pixel, welche nicht die Schnellste ist, werden optische Effekte nur bis zu einer gewissen Geschwindigkeit machbar und sind deshalb langsamer als die, die man bei individuellen Lösungen erreichen kann.



Vergrößern Diese Teile sind im Bausatz enthalten
© Swen Hopfe

Trotzdem fand ich es interessant, einen LED-Cube auf Basis eines solchen Bausatzes aufzubauen. Die Vorteile waren attraktiv genug für mich: Man stützt sich auf die Software von 4tronix und kann sich dann vor allem auf die eigene Kreativität bei den optischen Effekten konzentrieren. So entstand dieses Projekt.

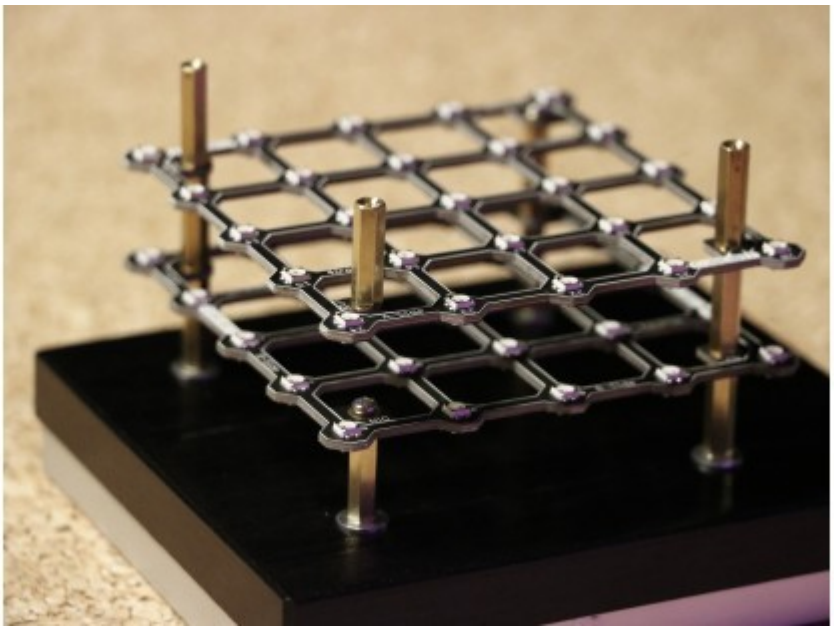
Den Bausatz gibt es in unterschiedlichen Größen. Um etwas mehr darstellen zu können, haben wir uns für die Gitter mit Kantenlänge von 5 LEDs entschieden. Geliefert wurde ordentlich verpackt. Auch die Abstandhalter gehören dazu.



Vergrößern Basisplatine: Der Unterboden des LED-Cubes
© Swen Hopfe

Mechanisch zu konstruieren war also wenig, aber es sollte ein ansehnlicher Grundkörper sein, in dem alle zusätzliche Elektronik untergebracht ist. Wir nutzen die für den Bausatz empfohlene Basisplatine. Die passt zu den LED-Elementen, stellt Anschlüsse für die

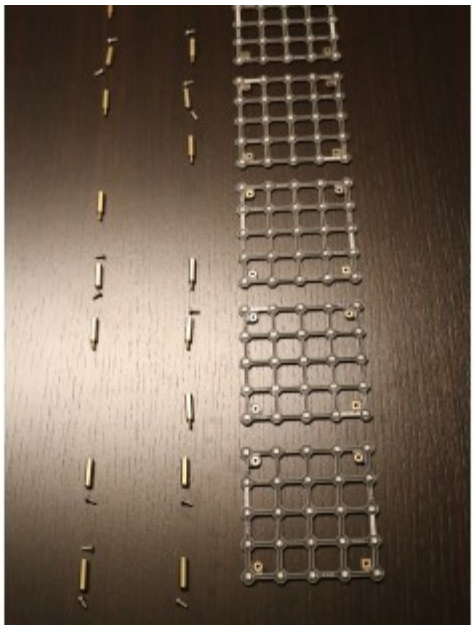
Stromversorgung bereit, und neben dem *micro:bit* lässt sich auch ein Raspberry Pi Zero stecken. Genau den wollen wir verwenden.



Vergrößern Schrittweiser Aufbau eines LED-Cubes
© Swen Hopfe

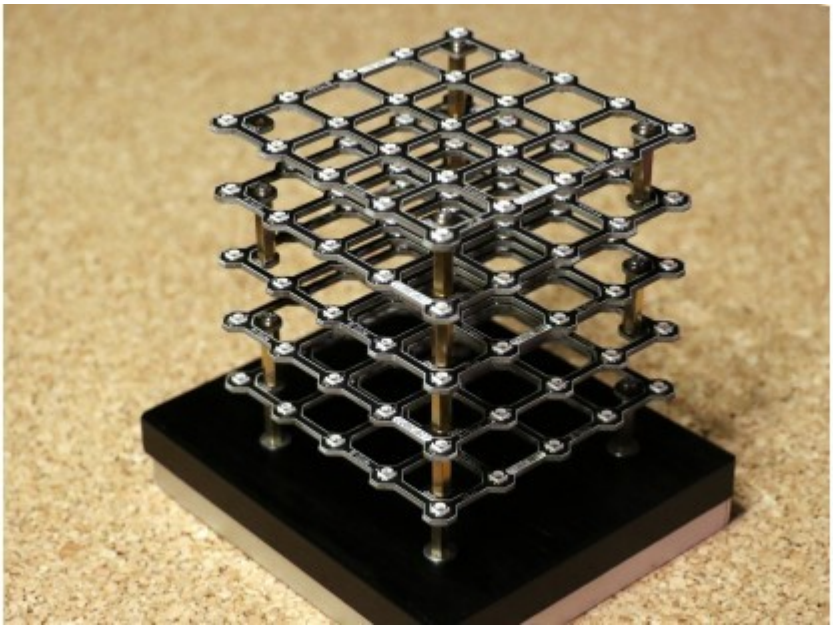
Jetzt muss der Cube Schritt für Schritt aufgebaut werden. Der 5x5x5-Cube benötigt fünf Slices, die mit Abstandhaltern miteinander verbunden werden. Jedes Element bekommt eine Spannungsversorgung, der Datenausgang eines Elements wird jeweils mit dem Dateneingang des darüberliegenden Nachfolgers verbunden.

Es gilt, die Anschlüsse nicht zu verwechseln und jeweils die richtige Seite nach oben zu drehen. Dafür sind die Slices günstigerweise mit „Side A“ und „Side B“ bereits herstellerseitig beschriftet. Am besten verschafft man sich vorher entsprechende Übersicht.



Vergrößern Schichtung der Module
© Swen Hopfe

Für den *micro:bit* wurde in Scratch programmiert, für den Pi ist die Neopixel-Library und entsprechende Software in Python verfügbar. Für die hier verwendeten Elemente gibt es einen Port *cubebit.py* , über den man die eigentlich strangförmig organisierten Neopixel in einem 3-Achsen-Koordinatensystem ansprechen kann. Das macht den Entwurf von Lichteffekten für den Würfel wesentlich einfacher. Mit diesem modularen „Bausatz“ an Soft- und Hardware kann man also richtig viel machen und die eigene Entwicklung integrieren.

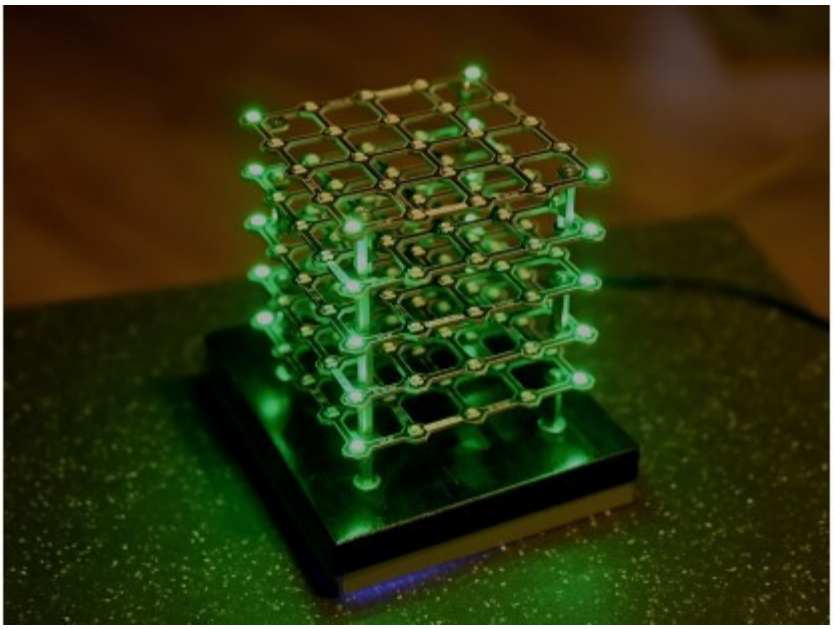


Vergrößern So sieht der fertige Cube aus
© Swen Hopfe

Unser eigenes Python-Demoscript zur Ansteuerung des Würfels ist auf Git-Hub unter https://github.com/swenae/neocube zu finden. Auf dem Steuer-Pi lassen wir es nach dem Einschalten starten.

Neben dem Bau und Test des Bausatzes war mir das Kennenlernen der *cubebit* -Library wichtig. Und da sollte eine kleine Weiterentwicklung auch nicht fehlen. So findet sich im Github-Projekt nicht nur die kleine Demo (*sw_cube.py*), sondern auch einige neue Methoden in *swc.py* , die den Umgang mit dem Cube vereinfachen sollen. Dort sind nun unter anderem „Lines“, „Slices“, „Squares“ und „Cubes“ festgelegt.

Die kann man nun nutzen, um schnell Linien, Rechtecke, Flächen oder „Subwürfel“ in einem Koordinatensystem innerhalb der Außengrenzen des Cubes zu erzeugen. Darauf basiert dann auch die Demo, die nicht nur die originalen Beispiele wie den Rainbow-Effekt integriert, sondern auch auf unsere Klasse mit den neuen Geometrien zugreift.



Vergrößern Muster auf dem LED-Cube
© Swen Hopfe

Vor dem Ausschalten kann man den Cube bzw. den Pi Zero über *ssh* an einem Terminal von einem Rechner im heimischen Netzwerk aus definiert herunterfahren. Da sich unser Pi im WLAN befindet, wird es auch möglich, auf unterschiedliche Ereignisse zu reagieren und diese anzeigen lassen. So könnte man eingegangene Mails melden oder ein Wettersymbol für die Vorschau auf den nächsten Tag anzeigen. Aber auch offline wird einiges möglich, etwa eine Anzeige für die Raumtemperatur, oder man reagiert mittels Mikrofon auf das Abspielen von Musik.

So soll unser Neo-Cube in nächster Zeit noch Bedienelemente an der Grundplatte erhalten, mit dem man ihn in den Ruhezustand versetzen und diverse Anzeigefunktionen schalten kann. In der Zwischenzeit erfreuen wir uns am „Showprogramm“, was zeigt, wie einfach und dekorativ Neopixel-Elemente für einen solchen LED-Würfel genutzt werden können.

Weitere ausgewählte interessante Projekte von Swen Hopfe:

- Eigenbau: Kamera-Motorstativ mit ESP32-Steuerung
- Smart-Home-fähig: Küchentimer im Eigenbau
- Raspberry Pi: Computerschach per Pi-Cam spielen
- NFC-Reader mit Raspberry Pi selbst bauen
- Raspberry Pi Zero W als smarter USB-Stick
- Zeitraffer-Aufnahmen mit dem Raspberry Pi Zero machen
- Webcam mit Raspberry Zero Pi W einrichten
- Nächtliche Tierfotografie mit NoIR-Cam und Raspberry Pi