

Eigener Treiber zur Ansteuerung der LED-Streifen

Im Netz gibt es verschiedene fertige Bibliotheken zur Ansteuerung der LED-Streifen. Alle bieten unterschiedliche Interfaces zur Programmierung an. Nicht alle erscheinen dabei maximal einfach. Außerdem betrachten sie häufig alle LEDs als Teil eines (linearen) Strangs, d.h. adressieren sie fortlaufend. Sind die Streifen aber matrixartig angeordnet (z.B. 15 Streifen à 10 LEDs zum Bau eines Tetris-Spiels), so wäre es einfacher, jede LED nur gemäß ihrer tatsächlichen Lage mit Zeilen- und Spaltenangabe adressieren zu können. Das ist insbesondere insofern interessant, als sich mehrere nebeneinandergeklebte Streifen auf unterschiedliche Weise verbinden lassen. Z.B. könnte man waagrecht kleben und jeweils abwechselnd mal links, mal rechts weiterverbinden (für zwei Beispiele mit 3 x 3 LEDs – für ein Tic-Tac-Toe-Spiel – siehe Abbildung 1). Bei konventioneller Adressierung ändert sich dabei jedesmal die Positionsangabe einer ausgewählten LED.

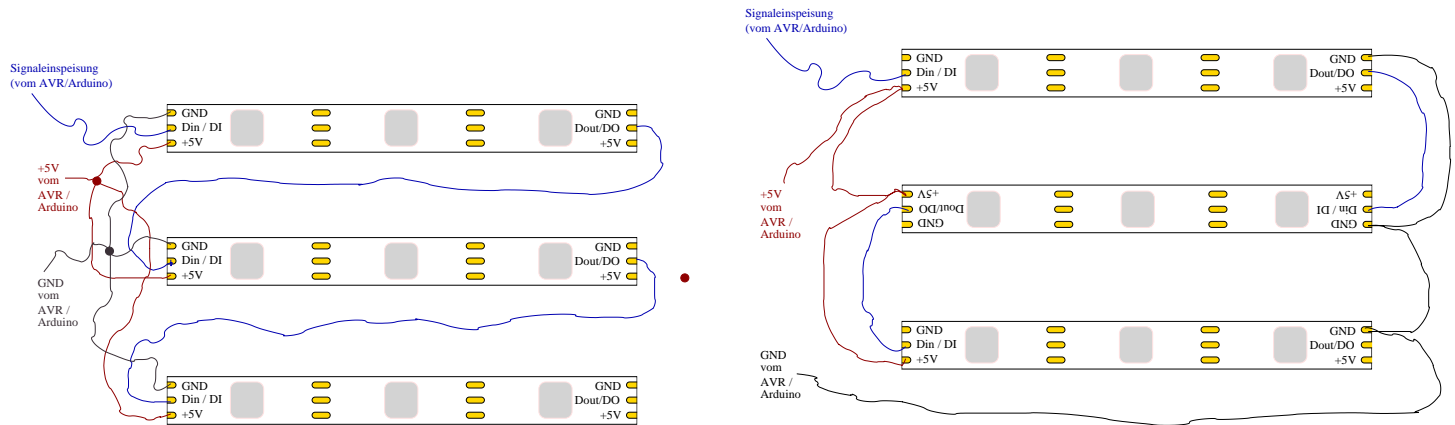


Abbildung 1: zwei Verdrahtungsbeispiele: 3x3-Matrix, Signaleinspeisung links oben, waagrechte Montage

Bei der linken Variante muss man vom Ende jedes Streifens (Dout) wieder zum gegenüberliegenden Anfang des nächsten Streifens (Din) zurückkehren. Die rechte Variante mit *gegenläufig geklebten Streifen* ist *einfacher zu verdrahten*. Hier kann man alle Drähte überschneidungsfrei verlegen und benötigt außerdem nur kurze Drahtstücke für die Signalleitungen. Dout eines Streifens und Din des nächsten Streifens liegen immer auf der jeweils gleichen Seite, also nicht mehr gegenüberliegend. Diese Verdrahtungsvariante benötigt die neue Bibliothek. In der Abbildung entspricht die linke Verdrahtungsvariante mit den gleichgerichteten Streifen dem Typ 5 und die rechte Verdrahtungsvariante mit den gegenläufigen Streifen dem Typ 4 (siehe umseitig). Beide haben ihre Einspeisung links oben.

Insgesamt sind 16 verschiedene Verdrahtungsmöglichkeiten für eine LED-Streifen-Matrix denkbar.¹ Damit der Aufbau einer Streifenschaltung nicht selbst schon zu einer Enttäuschung wird, wäre es hilfreich, wenn jedes *beliebige* gewählte Schema gleichermaßen einfach vom Treiber unterstützt würde. Außerdem wäre es hilfreich, wenn man bei der Ansteuerung entweder einzelne Werte für einzelne Farbkanäle setzen könnte (z.B. für “Rot”) oder aber Farben genau so, wie sie Testprogramme z.B. für Webseitengestaltung oder Bildverarbeitungsprogramme wie GIMP ausgeben, einsetzen könnte. Beide Aufgaben lassen sich nach meinen Recherchen mit den vorhandenen Bibliotheken nicht erfüllen (bei Raspberry Pi gar nicht und bei Arduinos nicht alle gleichzeitig).

Deshalb habe ich hier – auf der Grundlage der open-source-Bibliotheken `light_ws2812` und `ws2812-RPi` – eine eigene Bibliothek erstellt. Mit ihr kann ein Bastler Farbwerte aus Anwendungen übernehmen und er kann ein beliebiges Verdrahtungsschema auswählen und danach mit Zeilen- und Spaltenindex direkt auf die LEDs in ihrer Matrixanordnung zugreifen. Außerdem gibt es die neue Bibliothek für Raspberry Pi (`ws2812b_rpi`), für AVR-Controller (`ws2812b`) und für Arduino-Boards (`WS2812b_arduino`) gleichermaßen. Man kann also Quellcode gut austauschen.²

¹Man die Einspeisung (Din) entweder links oben oder links unten oder rechts oben oder rechts unten anbringen und die Streifen entweder kreuzweise verdrahten oder – kürzer – mit alternierender Laufrichtung.

²Das ist z.B. dann von Vorteil, wenn ein Prototyp am Raspberry Pi entwickelt wird, dann aber via AVR-Controller als eigenständiges Gerät weiterleben soll. Oder wenn sich während der Projektentwicklung mit Arduino herausstellt, dass dieser zu leistungsschwach oder speicherarm für die Aufgabe ist und besser durch z.B. einen Raspberry Pi Zero ersetzt würde.

Nachfolgend die 16 möglichen Verdrahtungsweisen einer LED-Matrix mit ihrer Typangabe in der Bibliothek. Am günstigsten zu verdrahten sind die Varianten mit gegenläufiger Klebung der Streifen (gerade Typennummer):

