

Teil 1

LEDs mit Intelligenz – RGB-LED mit integriertem Controller

Mit RGB-LEDs lassen sich vielfältige Lichtdekorationen, Anzeigen sowie LED-Werbe- und -Infoschilder aufbauen, jedoch ist der Aufwand für die Ansteuerung beträchtlich. Das änderte sich mit dem Erscheinen von RGB-LEDs mit integriertem Controller, denn hier ist der Treiber in jeder einzelnen LED integriert, weshalb nur noch Datenleitungen zur Ansteuerung nötig sind. Ein inzwischen weitverbreiteter Vertreter dieser LED-Klasse ist die WS2812B von Worldsemi. Wir wollen in einem zweiteiligen Beitrag deren Aufbau, Arbeitsweise und Anwendung betrachten und einen Bauvorschlag für eine 15x15-RGB-LED-Matrix auf Basis der WS2812B vorstellen.

Immer schön der Reihe nach ...

Die WS2812B [1] ist eine SMD-RGB-Leuchtdiode im 5050-Gehäuse, bei der sofort auffällt, dass sie nur über vier Anschlüsse (Bild 1) verfügt: Betriebsspannung, Masse, Data-in (DIN) und Data-out (DOU). Mehr braucht es hier auch nicht, denn in der LED ist neben den LED-Chips selbst ein LED-Controller untergebracht, der die Ansteuerung der LEDs übernimmt. Betrachtet man solch eine LED genau (Bild 2), kann man die einzelnen Bestandteile gut mit bloßem Auge erkennen. Als Controller kommt der auch als einzelner DIP-Chip erhältliche WS2811 [2] zum Einsatz. Er enthält einen programmierbaren Konstantstrom-Treiber, einen internen Präzisionsoszillator, einen Signalformer und für jeden Farbkanal ein 8-Bit-PWM-Register, in das je 256 Helligkeitswerte geladen werden. Ist das Laden der insgesamt 24 Bit (8 Bit Grün, 8 Bit Rot, 8 Bit Blau) in das Schie-

beregister abgeschlossen, erfolgt durch einen Reset die Übernahme der Daten in das Register und danach ein Weiterschieben der nächsten Daten über den Datenausgang zum nächsten Chip in der LED-Anordnung (Daisy-Chain). Die Ansteuerung erfolgt über nur eine Datenleitung mit einem asynchronen, seriellen Protokoll (NZR). Bild 3 illustriert den Ablauf am Beispiel mit drei Controllern.

Abweichend von üblichen Logikschemen wird eine logische Null mit einem kurzen High-Impuls, eine logische Eins mit einem längeren High-Impuls erzeugt,

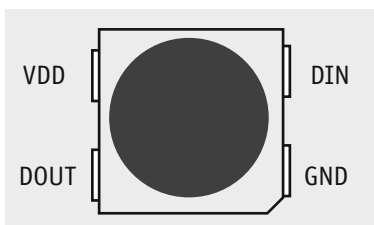


Bild 1: Die Anschlussbelegung der WS2812B



Bild 2: In der Nahansicht erkennt man deutlich die LED-Chips und den Controller auf dem Substrat.

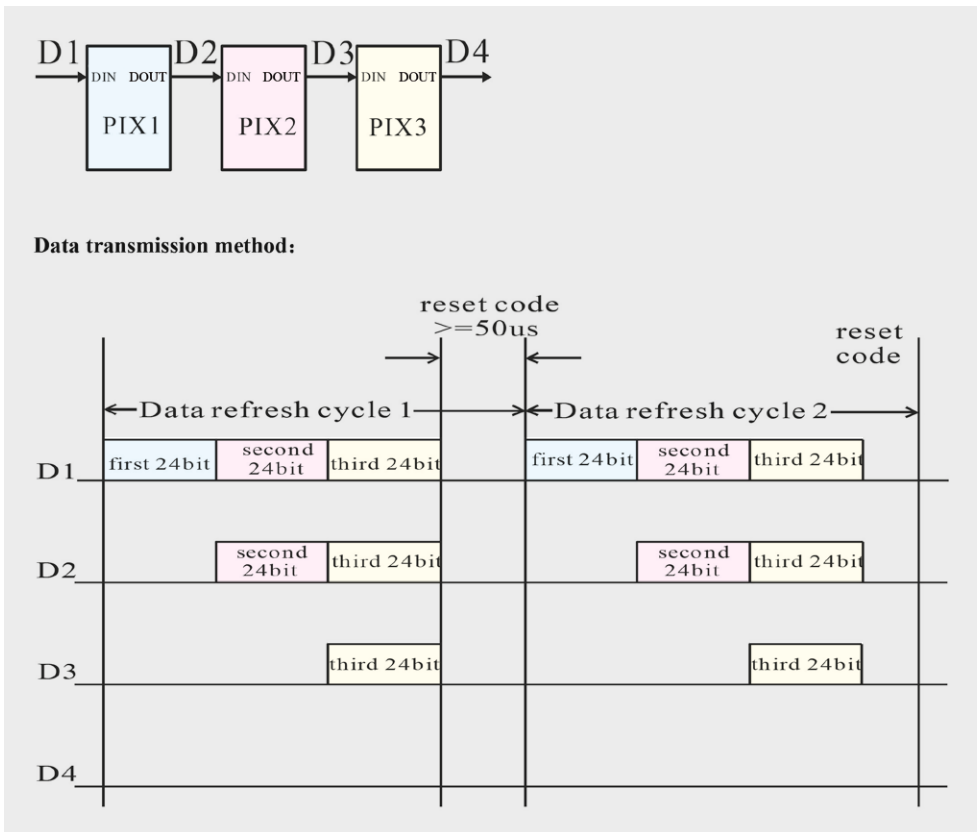


Bild 3: Das Datentransferschema beim Durchschieben der Daten in der LED-Kette

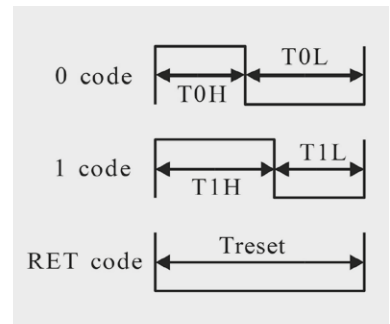


Bild 4: Das Logikschema für den seriellen Datentransfer zur WS2812B

der Reset-Code wird mit einem 50 µs langen Low-Impuls übermittelt (Bild 4). Tabelle 1 zeigt das Zeitregime, dabei sieht man, dass es hier um recht schnelle Zeitabläufe geht, entsprechend schnell müssen also die Programmlaufzeiten innerhalb eines Ansteuerungsprogramms sein, weshalb die meisten Software-Bibliotheken auch auf dem kompakten Bit-Banging-Prinzip beruhen, so etwa die weitverbreitete Adafruit Neopixel Library [3], die Pololu-LED-Strip-Library oder FastSPI_LED Library. Besonders die Adafruit Neopixel Library wird gern für eigene Ansteuerprogramme genutzt, sie ist kompakt, einfach einsetzbar und universell. Die Bezeichnung „Neopixel“ ist übrigens eine weitverbreitete Bezeichnung für die Worldsemi-RGB-LEDs mit serieller Ansteuerung.

Wichtig bei der Programmierung von Ansteuerprogrammen ist ein präzises Timing, bereits recht geringe Timingabweichungen führen zu Ausgabefehlern. Wer allerdings lediglich vorhandene Programme nutzt, muss sich hier nicht um Timingprobleme kümmern. Aus Anwendersicht gilt es hier lediglich, die Vorgaben der einzelnen Programme zu beachten, wie z. B. die maximale Anzahl der LEDs (Pixel). Auf diese ist das Timing ausgelegt. Auch der etwas versteckte Hinweis im Datenblatt der LED („Follow the order of GRB to sent data and the high bit sent at first“) ist zu beachten: Das Ausgeben der einzelnen Bytes erfolgt nicht in der aus der Sprachpraxis gewohnten Reihenfolge RGB, sondern GRB. Das muss man bei der Konfiguration von Ansteuerprogrammen beachten.

AVR, Arduino & Co.

Die kompakten Programme und die Bibliotheken zur Ansteuerung der LEDs implizieren natürlich den Einsatz von kleinen Mikroprozessorslösungen für die Codeerzeugung bzw. -umsetzung. Da wir hier keine Leistungstreiber benötigen, genügt eine Datenleitung zur Ansteuerung. So kann ein kleiner AVR oder typischerweise ein Arduino oder ein kleiner ARM-Rechner wie der Raspberry Pi die Ansteuerung übernehmen.

Besonders die Arduino-Plattform ist wegen der weiten Verbreitung und ihres Auftretens als bereits komplettes Mikrocontrollersystem enorm beliebt. Dabei agieren die Mikrocontroller entweder als Stand-

Tabelle 1: Datentransferzeiten (TH+TL=1,25 µs+600 ns)

Tabelle 1	T0H	0 code, High-Zeit	0,4 µs*150ns
	T1H	1 code, High-Zeit	0,8 µs*150ns
	T0L	0 code, Low-Zeit	0,85 µs*150ns
	T1L	1 code, Low-Zeit	0,45 µs*150ns
	RES	RET code, Low-Zeit	> 50 µs

alone-Systeme mit eigenen geladenen Anwendungsprogrammen oder lediglich als Codekonverter, die per Netzwerk/USB empfangene oder per SD-Karte eingespielte Anzeigesequenzen, die mit den verschiedensten Protokollen erzeugt werden können, in die seriellen Steuercodes für die WS2812B umsetzen. Bild 5 zeigt einen Vertreter der zuletzt aufgeführten Controller, einen typischen Playerbaustein, der autark von einer microSD-Karte liest oder per USB von einem PC-Programm angesteuert wird.

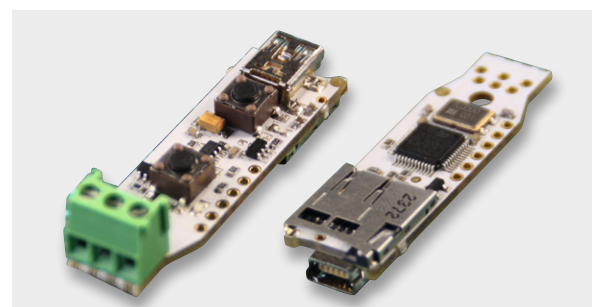


Bild 5: Typischer Protokoll-Konverter – der DiameX-LED-Controller kann sowohl Daten direkt per USB empfangen und umsetzen als auch Dateien im tpm2-Format von einer microSD-Karte abspielen.

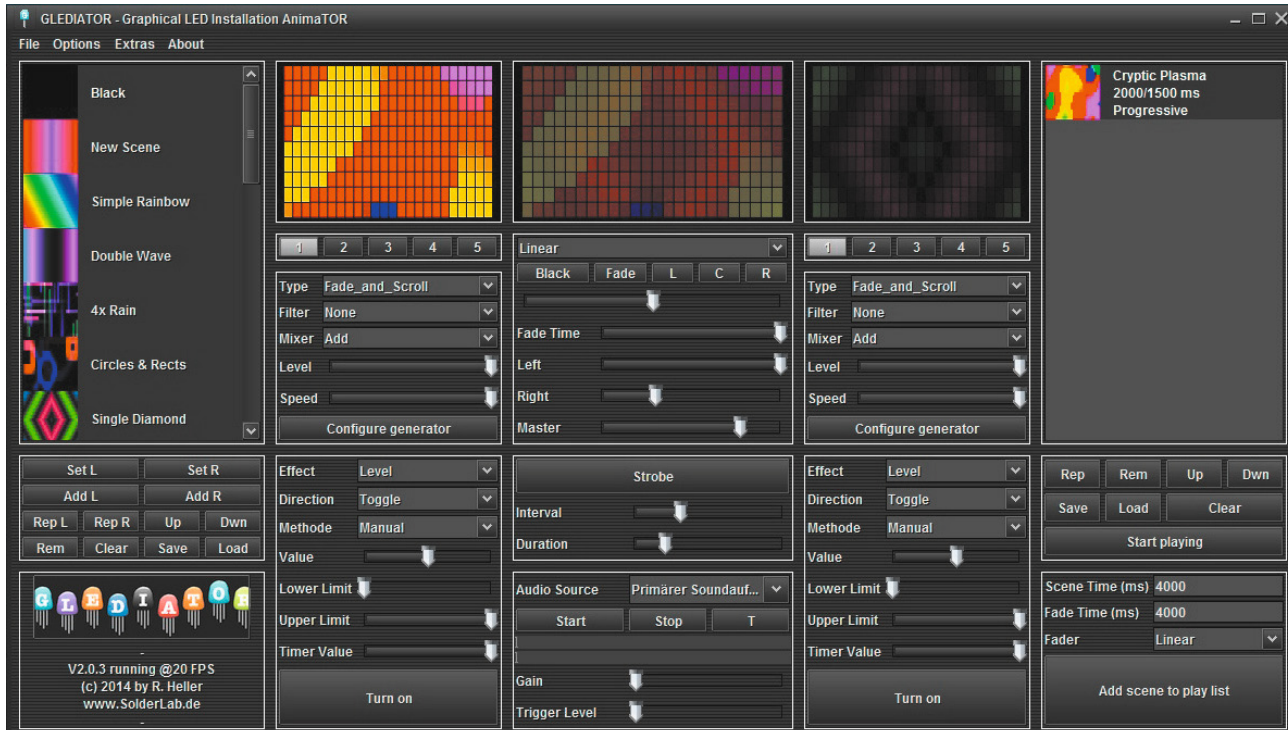


Bild 6: Tolle Arbeit von René Heller: das Matrix-Controller-Programm Glediator

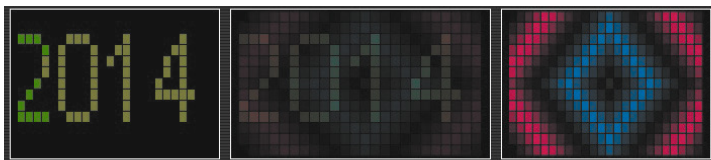


Bild 7: Auch (Lauf-)Texte können in die Effekte eingebunden werden.

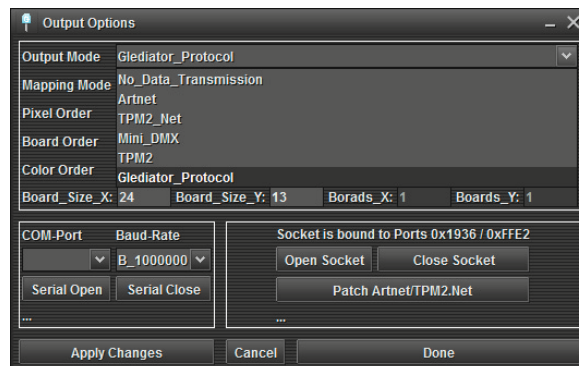


Bild 8: Die Ausgabe-Auswahl von Glediator bietet alle gängigen Protokolle.

COM, ARTNET, TPM2, TPM2.net, Mini-DMX?

Will man eine WS2812B-Anordnung nicht nur als festen Lichteffect, statische Anzeige o. Ä. einsetzen, greift man zu einem LED-Matrix-Kontrollprogramm. Die wohl beliebtesten Programme sind die Freeware-Programme Glediator von René Heller [4], Jinx! von Sven Karschewski [5] und PixelController von Michael Vogt [6].

Alle drei funktionieren ähnlich, in mehreren Effektkanälen kann man vorhandene bzw. eigene Pixelmuster, Texte, Farbverläufe etc. auswählen bzw. kreieren, in einem Summenkanal mischen, dabei u. a. auch von Soundfiles steuern lassen und so unendlich viele Muster, Effekte, Textausgaben usw. selbst erzeugen. Bild 6 zeigt als Beispiel die Oberfläche von Glediator, Bild 7 eine Mischsequenz aus Text und Effekt. Praktischerweise bietet die Glediator-Internetseite gleich noch auf allen Arduino-Plattformen einsetzbare Konvertierprogramme, die das programmeigene Glediator-Protokoll in das beschriebene serielle Protokoll umsetzen. Ausgabeprotokolle gibt es einige (Bild 8 zeigt die Auswahl von Glediator), so verwenden quasi alle Controller-Programme das von einigen Mitgliedern der LEDSTYLES-Community [7] kreierte Protokoll tpm2, das es für die Ausgabe via USB ebenso gibt wie via Ethernet (tpm2.net). Ziel der drei Entwickler war es, quasi beliebige USB-/Seriell-Adapter einsetzen zu können, ohne an eine bestimmte Hardware gebunden zu sein. Entsprechend beherrschen alle gängigen Umsetzer/Player,

wie z. B. auch der in Bild 5 gezeigte USB-/SD-Card-Player, dieses Protokoll. Eine detaillierte Einführung zu tpm2 findet sich unter [7].

Mit Mini-DMX und ArtNet finden sich auch aus der professionellen Lichtsteuerungswelt entlehnte bzw. dort genutzte DMX-Protokolle, sodass auch LED-Lichtausgabegeräte mit DMX-Interface ansteuerbar sind. Denn längst findet man die WS2812B auch in der Profi-Welt.

WS2812B in der Praxis

Die WS2812B sind als Einzel-LEDs oder, wie in Bild 9 zu sehen, fertig konfektioniert auf selbstklebendem, flexiblem Trägerband erhältlich. Besonders die letztere, insgesamt auch preiswertere Version ist sehr beliebt. Dabei kann der Streifen nach jeder LED aufgetrennt werden. So kann man sich schnell und unkompliziert eine komplette Matrix (Bild 10) bauen.

Im praktischen Umgang mit der WS2812B kommt es immer wieder zu Ausfällen der ersten LED einer Anordnung, nämlich genau dann, wenn die Betriebsspannung geringer ist als die auf der Datenleitung.



Denn die Grenzen sind hier laut Datenblatt eng gesetzt. Das kann z. B. passieren, wenn man die Betriebsspannung mitten im Betrieb abschaltet oder das Netzteil zu schwach ist, um unter allen Umständen (immerhin benötigt jede LED je nach Helligkeit und Farbe bis zu 60 mA) tatsächlich die benötigten 5 V (mindestens 3,5 V) zu liefern. Deshalb haben Anwender und auch inzwischen kommerzielle Anbieter Schutzmaßnahmen entwickelt, die man z. B. auch zahlreich als Hinweis in den Arduino-Sketches von Adafruit findet.

Zunächst also sollte die Stromversorgung ausreichend bemessen sein. Ein üblicher 4-m-Streifen mit 240 LEDs benötigt immerhin 14,4 A! Bei derartigen Strömen sollte man bei größeren Anordnungen nicht nur ausreichende Leitungsquerschnitte der Spannungsversorgungsleitungen wählen, sondern auch die Einspeisung mehrfach, z. B. für jede Zeile einer Matrix, installieren. Auch bei der Spannungsweiterleitung zwischen Streifenabschnitten sind ausreichend starke Leitungen einzusetzen.

Der unbedingt laut Datenblatt nötige 100-nF-Kondensator an jeder LED (Bild 11) ist auf den meisten LED-Streifen bereits vorhanden, wenn nicht, dann muss er unmittelbar an den Betriebsspannungsanschlüssen installiert werden. Zusätzlich empfiehlt z. B. Adafruit, jeden Einspeisepunkt mit einem 1000-µF-Elko zu puffern.



Bild 9: Leicht zu verarbeiten, nach jeder LED teilbar: ein WS2812B-LED-Streifen

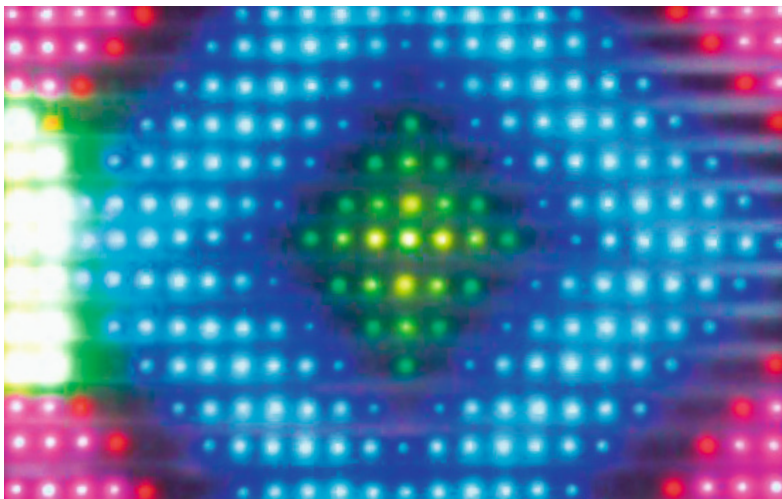


Bild 10: Die selbstklebenden LED-Streifen machen einen Matrix-Selbstbau einfach. Hier eine 24x13-Matrix, zur besseren Lichtverteilung in einer Doppelstegplatte eingebaut

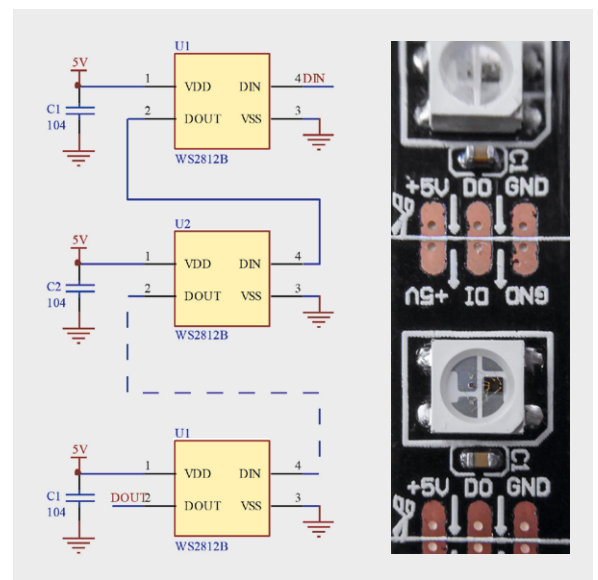


Bild 11: Die Abblockkondensatoren sind auf den konfektionierten Streifen bereits vorhanden.



Weitere Infos:

- [1] Datenblatt WS2812B
www.world-semi.com/en/Driver/Lighting/WS2811/WS212B/
- [2] Datenblatt WS8211
www.world-semi.com/en/Driver/Lighting_LED_driver_chip/WS2811/
- [3] Adafruit Neopixel Library
https://github.com/adafruit/Adafruit_NeoPixel
- [4] Glediator (Matrix-Controller-Programm)
www.solderlab.de
- [5] Jinx! (Matrix-Controller-Programm)
www.live-leds.de/
- [6] PixelController (Programm)
http://pixelinvaders.ch/?page_id=160
- [7] tpm2-Protokoll
www.ledstyles.de/ftopic18969.html
- [8] Schutzschaltung
www.led-studien.de/2014/06/24/schutzschaltung-fuer-ws2812-pixel/

Zum Schutz gegen zu hohe Spannungen und Störungen auf der Signalleitung werden folgende Maßnahmen empfohlen: möglichst kurze Leitungslänge zum Controllerausgang, ggf. sogar geschirmt, und ein Serienwiderstand (100-200 Ω) vor der ersten LED der Matrix. Einen erweiterten Schutz gegen Überspannungen auf Stromversorgungs- und Signalleitung bietet eine Suppressor-/Z-Dioden-Anordnung, wie man sie z. B. unter [8] findet.

Schließlich versteht es sich von selbst, dass man die LED nicht bei auf der Stromversorgungsleitung liegender Versorgungsspannung anschließt, sondern nur bei abgeschalteter Spannung. Zur Sicherheit immer zuerst die Masseleitung anschließen!

Beachtet man diese wenigen einfachen Hinweise, wird man lange Freude an seiner flexibel einsetzbaren LED-Matrix oder anderen Effekten mit der WS2812B haben!

Im zweiten Teil dieses Beitrags werden wir den Aufbau einer 15x15-Pixel-Matrix und die Ansteuerung über einen USB-/SD-Karten-Controller und Jinx! beschreiben.

