

HSV-Farben im Pattern Configurator



Motiviert durch den komplizierten [Farbwechsel](#) meiner Burg wollte ich schon immer verstehen, wie der viel einfachere Farbwechsel mit HSV-Farben umzusetzen ist. Dank Norberts Hilfe beim Oktober-Stammtisch 2023 konnte ich das Rätsel nun endlich knacken und lasse euch wie im Forum versprochen daran teilhaben.

— [Michael](#) 2023/11/07

Was sind die HSV-Farben?

Der HSV-Farbraum beschreibt den Ort einer Farbe innerhalb eines Kegels. Zum besseren Grundverständnis eignet sich der entsprechende [Artikel auf Wikipedia](#).

- **H**ue (Farbwert): Farbwinkel innerhalb des Farbkreises. Rot (0°), Grün (120°), Blau (240°) und am Ende wieder Rot (360°)
- **S**aturation (Sättigung): Neutralgrau (0%), wenig gesättigte Farbe (50%), gesättigte bzw. reine Farbe (100%)
- **V**alue (Helligkeit): keine Helligkeit (0%), volle Helligkeit (100%)

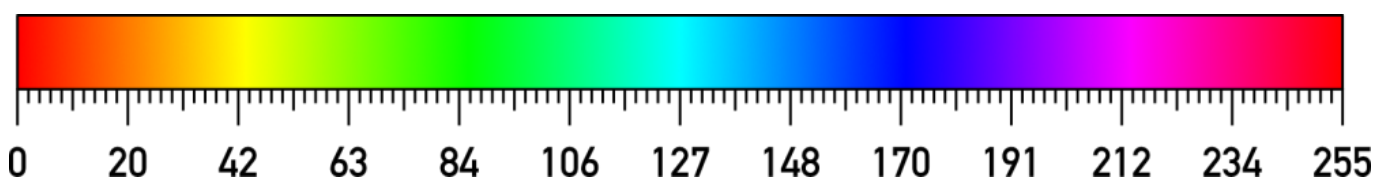
Da bei der MobaLedLib keine Werte in Prozent eingeben werden können, wird dieses Farbsystem mit den 8bit (0-255) umgesetzt, die die FastLED Bibliothek* zur Verfügung stellt. Daraus ergeben sich dann folgende Werte:

- **H**ue (Farbwert): Farbwinkel innerhalb des Farbkreises. Rot (0), Grün (74), Blau (170) und am Ende wieder Rot (255)
- **S**aturation (Sättigung): Neutralgrau (0), wenig gesättigte Farbe (127), gesättigte bzw. reine Farbe (255)
- **V**alue (Helligkeit): keine Helligkeit (0), volle Helligkeit (255)

*) Link zur FastLED-Beschreibung: [FastLED-HSV-Colors](#)

Der wichtigste Wert ist selbstverständlich der Farbwert selbst (Hue), denn Helligkeit und Sättigung beeinflussen jeden Farbwert in gleichem Maße.

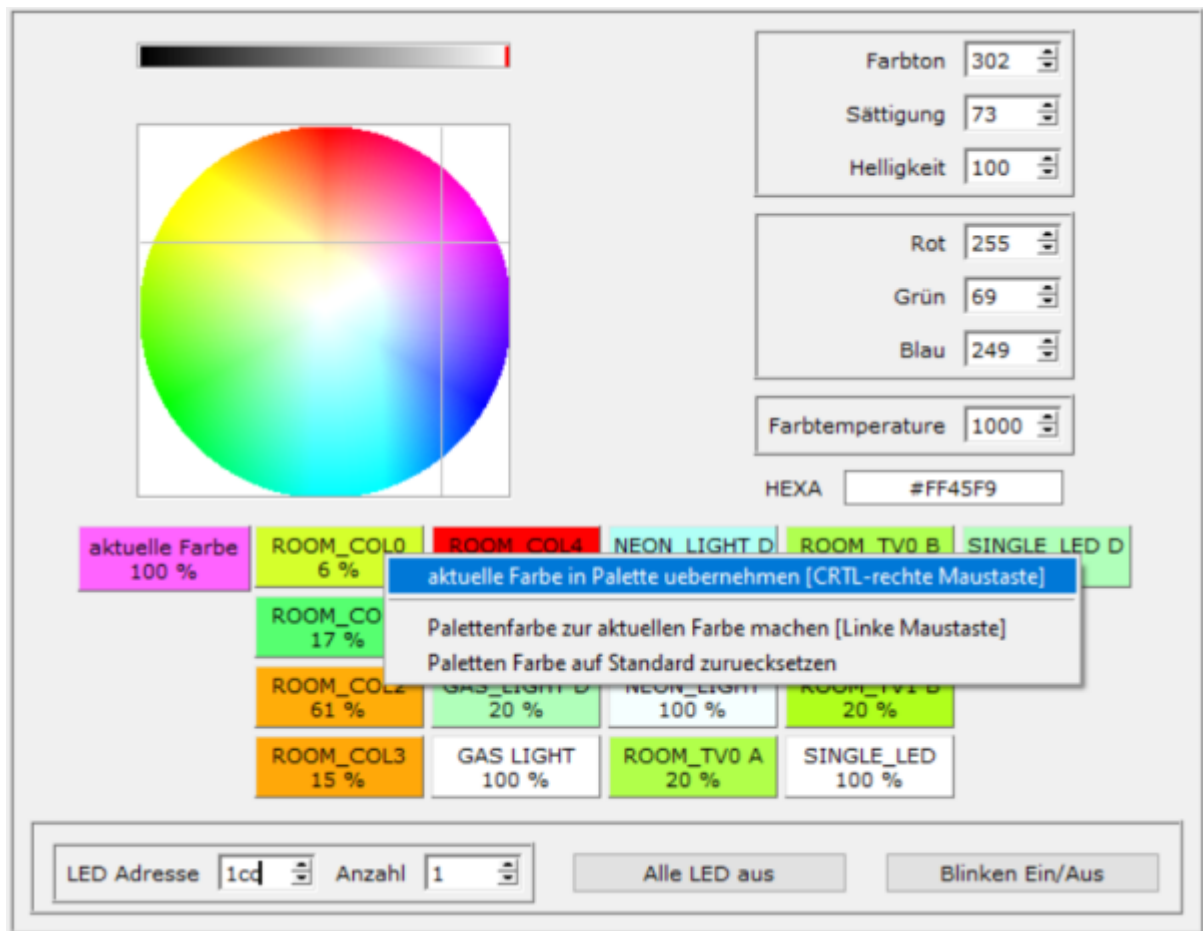
Der folgende Verlauf zeigt, mit welchem Wert eine bestimmte Farbe zu erzielen ist. Für alle im Verlauf dargestellten Farben sind sowohl Sättigung als auch Helligkeit auf 255 eingestellt.



Mithilfe der HSV-Werte lassen sich alle Farben für eine RGB-LED sehr schnell definieren. Eine gute Hilfe bietet dabei das Farbtestprogramm von Harold.

Hier kann man Farbe und Sättigung durch Verschieben des Fadenkreuzes und die Helligkeit mithilfe des Reglers über dem Farbkreis einstellen. Die drei Werte kann der Pattern Configurator im Mode

PM_HSV dann direkt interpretieren.



Wozu braucht man das?

Gerade Farbverläufe, wie sie heute vielerorts zur Effektbeleuchtung eingesetzt werden, belegen mit RGB Farben etwas mehr Speicher auf dem Arduino. Zudem ist das ständige Hantieren mit den ganzen RGB-Werten (jeweils drei pro Farbe) viel umständlicher als mit einem Hue-Wert.

Ein ganz besonderer Vorteil ist es beispielsweise, sechs Einzel-LEDs im Pattern Configurator anzulegen und diese in sechs Zeilen des Programm Generators zu kopieren. Will man beispielsweise 18 LEDs damit ansteuern, kann man die jeweiligen Farben mit dem „Copy LED“-Befehl vervielfachen.

Wie programmiert man das?

Nur Farbwert einstellen

Wenn man nur die Farbwerte programmieren möchte (Hue), kann man im Pattern Configurator etwas Speicher sparen (2), indem man nur einen Ausgabekanal angibt (1). In dem Fall steuert man mit dem ersten Ausgabekanal nicht wie sonst üblich den roten Chip auf einer WS2812, sondern den Farbwert aller drei Chips. Die fehlenden Ausgangskanäle Saturation und Value werden bei nur einem Ausgabekanal automatisch auf 255 (100%) gesetzt. Konkret sieht das so aus:

Ver.: 3.2.1 28.12.22 Neues Blatt by Hardi

Erste RGB LED: 1
 Startkanal der RGB LED: 0
 Schalter Nummer: SI_1
 Anzahl der Ausgabe Kanäle: 1 1
 Bits pro Wert: 8 1 => 256 Helligkeitsstufen (0..255)
 Wert Min: 0
 Wert Max: 255
 Wert ausgeschaltet: 0
 Mode: PM_HSV
 Analoges Überblenden: 1
 Goto Mode: 0
 Grafische Anzeige: 1
 Spezial Mode:

Aktualisieren Test Pattern

Ergebnis: **APatternT4(1,28,SI_1,1,0,255,0,PM_HSV,0,6 Sek,0,6 Sek,0,127,127,255)**

Makro Name: **HSV_rot**

Makro: **#define HSV_rot(LED,InCh) APatternT4(LED,28,InCh,1,0,;**
#define HSV_rot_StCh(LED,StCh,InCh) APatternT4(LED,StCh+28,InCh,1

2 Wenn gleiche Zeiten verwendet werden, dann sollten nur die ersten Zeiten

Dauer	0	6 Sek	0	6 Sek		
Flash Bedarf: 23 Bytes						

☐ RGB LED

LED Nr	Spalte Nr ->	1	2	3	4	5	6
1	Hue	0	127	127	255		

Import von Prog. Gen. Programm Generator

Alle Werte einstellen

Will man den Farbwert (Hue) und die Sättigung (Saturation) einstellen, wählt man zwei Ausgabekanäle, will man alles einstellen, wählt man alle drei Ausgabekanäle (1). Eine Besonderheit stellt die Steuerung von Farbwert (Hue) und Helligkeit (Value) dar. Wer nur diese beiden Werte einstellen will, muss alle drei Ausgabekanäle auswählen, weil die Sättigung (Saturation) immer an Stelle zwei steht. In dem Fall muss bei der Sättigung auch ein Wert eingetragen werden (3), denn wenn die Zeile leer bleibt, geht das Programm vom Wert 0 aus. Die LED wäre also aus.

Neues Blatt

Erste RGB LED:	1
Startkanal der RGB LED:	0
Schalter Nummer:	SI_1
Anzahl der Ausgabe Kanäle:	3
Bits pro Wert:	8
Wert Min:	0
Wert Max:	255
Wert ausgeschaltet:	0
Mode:	PM_HSV
Analoges Überblenden:	1
Goto Mode:	0
Grafische Anzeige:	1
Spezial Mode:	

=> 256 Helligkeitsstufen (0..255)

Aktualisieren

Test Pattern

Makro Name: HSV_rot

```
Makro: #define HSV_rot(LED,InCh)          APatternT4(LED,28,InCh,3,0,;
```

```
#define HSV_rot_StCh(LED,StCh,InCh)  APatternT4(LED,StCh+28,InCh,3
```

Wenn gleiche Zeiten verwendet werden, dann sollten nur die ersten Zeiten

Dauer	0	6 Sek	0	6 Sek		
-------	---	-------	---	-------	--	--

Flash Bedarf: 31 Bytes

+	-	<input type="checkbox"/> RGB LED
---	---	----------------------------------

LED Nr	Spalte Nr ->	1	2	3	4	5	6
1	Hue	0	127	127	255	3	
2	Saturation	x	x	x	x		
3	Value	x	x	x	x		

Der Vorteil von Variante 1 liegt in der Einfachheit und beim Speicherbedarf der Programmierung. Schwierig wird es, wenn man diese Variante beispielsweise mit einer Goto-Tabelle ein- und ausschalten will.

Ablauf des Farbwechsels

<https://wiki.mobaledlib.de/>

beispielsweise 0 nach 255 in zehn Sekunden lässt man die Farbe von 255 nach 0 in null Sekunden wechseln. Der Farbwechsel rotiert.

Da in dem Fall die Farben 0 (rot) und 255 (rot) annähernd identisch sind, kann das Auge diesen Sprung nicht erfassen. Dasselbe gilt selbstverständlich für einen Wechsel von 43 (gelb) über 255 (rot) nach 42 (gelb). Damit der Verlauf im Anschluss nicht über 255 zurück nach 43 wandert, definiert man diesen Vorgang mit null Sekunden.

Beispiel mit identischer Sättigung und Helligkeit:

Im Folgenden sind die sechs Grundfarben Rot (0), Gelb (42), Grün (84), Cyan (127), Blau (169) und Magenta (211) als Farbwechsel mit jeweils zwei Sekunden pro Farbton und einer Wechseldauer von 12 Sekunden je Durchgang dargestellt.



Das hier gezeigte Prinzip stellt alle Farben mit gleicher Sättigung und mit gleicher Helligkeit über den gesamten Farbwechsel dar. Will man die verschiedenen Farben in ihrer Helligkeit beeinflussen, muss man sie im Pattern Configurator vereinzeln. So kann man einen Ausgleich zwischen intensivem Rot und schwachem Blau schaffen. Ein Beispiel dazu folgt weiter unten.

Rotierender Farbwechsel beginnend mit Rot (255):

Dauer		12 Sek	0
Flash Bedarf: 21 Bytes			
+ - <input type="checkbox"/> RGB LED			
LED Nr	Spalte Nr ->	1	2
1	Hue	255	0
2	Saturation	x	x
3	Value	127	127

Rotierender Farbwechsel beginnend mit Gelb (42):

Dauer		10 Sek	0	2 Sek
Flash Bedarf: 26 Bytes				
+ - <input type="checkbox"/> RGB LED				
LED Nr	Spalte Nr ->	1	2	3
1	Hue	255	0	42
2	Saturation	x	x	x
3	Value	127	127	127

Rotierender Farbwechsel beginnend mit Grün (84):

Dauer		8 Sek	0	4 Sek
Flash Bedarf: 26 Bytes				
+ - <input type="checkbox"/> RGB LED				
LED Nr	Spalte Nr ->	1	2	3
1	Hue	255	0	84
2	Saturation	x	x	x
3	Value	127	127	127

Rotierender Farbwechsel beginnend mit Cyan (127):

Dauer		6 Sek	0	6 Sek
Flash Bedarf: 26 Bytes				
+ - <input type="checkbox"/> RGB LED				
LED Nr	Spalte Nr ->	1	2	3
1	Hue	255	0	127
2	Saturation	x	x	x
3	Value	127	127	127

Rotierender Farbwechsel beginnend mit Blau (169):

Dauer		4 Sek	0	8 Sek
Flash Bedarf: 26 Bytes				
+ - <input type="checkbox"/> RGB LED				
LED Nr	Spalte Nr ->	1	2	3
1	Hue	255	0	169
2	Saturation	x	x	x
3	Value	127	127	127

Rotierender Farbwechsel beginnend mit Magenta (211):

Dauer		2 Sek	0	10 Sek
Flash Bedarf: 26 Bytes				
+ - <input type="checkbox"/> RGB LED				
LED Nr	Spalte Nr ->	1	2	3
1	Hue	255	0	211
2	Saturation	x	x	x
3	Value	127	127	127

Beispiel mit unterschiedlicher Helligkeit:

Im Folgenden sind die sechs Grundfarben Rot (0), Gelb (42), Grün (84), Cyan (127), Blau (169) und Magenta (211) als Farbwechsel mit jeweils zwei Sekunden pro Farbton und einer Wechseldauer von 12 Sekunden je Durchgang dargestellt.

Richtung des Farbwechsels:

Im Programm Generator werden nun die sechs Pattern untereinander mit derselben Adresse angesteuert. Beginnt man mit Rot, so läuft der Farbwechsel von rechts nach links.

✓	1	AnAus	1	HSV_rot (pc)	Speicher für HSV reservieren	New HSV_Group()		
✓	1	AnAus	1	HSV_gelb (pc)	Muster Pattern_Configurator	APatternT2(#LED,28,#InCh,3,0,255,0,PM HSV,12 Sek,0	1	1
✓	1	AnAus	1	HSV_gruen (pc)	Muster Pattern_Configurator	APatternT3(#LED,28,#InCh,3,0,255,0,PM HSV,10 Sek,0	2	1
✓	1	AnAus	1	HSV_cyan (pc)	Muster Pattern_Configurator	APatternT3(#LED,28,#InCh,3,0,255,0,PM HSV,8 Sek,0	3	1
✓	1	AnAus	1	HSV_blau (pc)	Muster Pattern_Configurator	APatternT3(#LED,28,#InCh,3,0,255,0,PM HSV,6 Sek,0	4	1
✓	1	AnAus	1	HSV_magenta (pc)	Muster Pattern_Configurator	APatternT3(#LED,28,#InCh,3,0,255,0,PM HSV,4 Sek,0	5	1
✓	1	AnAus	1		Muster Pattern_Configurator	APatternT3(#LED,28,#InCh,3,0,255,0,PM HSV,2 Sek,0	6	1

Stellt man Rot ans Ende, so läuft der Farbwechsel von links nach rechts.

✓	1	AnAus	1	HSV_magenta (pc)	Speicher für HSV reservieren	New HSV_Group()		
✓	1	AnAus	1	HSV_blau (pc)	Muster Pattern_Configurator	APatternT3(#LED,28,#InCh,3,0,255,0,PM HSV,2 Sek,0	1	1
✓	1	AnAus	1	HSV_cyan (pc)	Muster Pattern_Configurator	APatternT3(#LED,28,#InCh,3,0,255,0,PM HSV,4 Sek,0	2	1
✓	1	AnAus	1	HSV_gruen (pc)	Muster Pattern_Configurator	APatternT3(#LED,28,#InCh,3,0,255,0,PM HSV,6 Sek,0	3	1
✓	1	AnAus	1	HSV_gelb (pc)	Muster Pattern_Configurator	APatternT3(#LED,28,#InCh,3,0,255,0,PM HSV,8 Sek,0	4	1
✓	1	AnAus	1	HSV_rot (pc)	Muster Pattern_Configurator	APatternT3(#LED,28,#InCh,3,0,255,0,PM HSV,10 Sek,0	5	1
✓	1	AnAus	1		Muster Pattern_Configurator	APatternT2(#LED,28,#InCh,3,0,255,0,PM HSV,12 Sek,0	6	1

Farbwechsel von der Mitte aus in beide Richtungen:

Soll der Farbwechsel beispielsweise auf elf LEDs verteilt werden, bei der die Farbe von der mittleren in beide Richtungen nach außen läuft, so kann man den Copy-LED Befehl nutzen.

Dazu nimmt man den Verlauf von rechts nach links und setzt die Copy-Befehle in umgekehrter Reihenfolge drunter. Selbstverständlich lassen sich „Copy-LED“-Befehle auch zwischen die einzelnen

