

Dokumentation zum

PitschuLight

Ein Full-Screen AmbiLight mit bis zu 312 LEDs auf Basis des TVP5150AM1
Video Prozessors und eines STM32F4discovery boards

Stand: Version 1.2, November 2014



Das Foto stammt von Pat Wat (frankp)

So sieht's aus wenn es funktioniert

Disclaimer

/******

*
* Control program for the PitschuLight TV-Backlight
* (c) Peter Schulten, Mülheim, Germany
* peter_(at)_pitschu.de
*

* Die unveränderte Wiedergabe und Verteilung dieses gesamten Sourcecodes
* in beliebiger Form ist gestattet, sofern obiger Hinweis erhalten bleibt.
*

* Ich stelle diesen Sourcecode + Doku kostenlos zur Verfügung und biete daher weder
* Support an noch garantiere ich für seine Funktionsfähigkeit. Außerdem
* übernehme ich keine Haftung für die Folgen seiner Nutzung.
*

* Der Sourcecode darf nur zu privaten Zwecken verwendet und modifiziert werden.
* Darüber hinaus gehende Verwendung bedarf meiner Zustimmung.
*

* Siehe auch: <http://www.mikrocontroller.net/topic/302189>
* History

* 09.06.2013 pitschu Start of work
* 19.11.2013 pitschu first release
* 05.05.2014 pitschu v1.1 supports blue user button (mode switch/standby)
* 24.07.2014 pitschu v1.2 bug fixing in tvp5150_dcmi.c
*/

Änderungshistorie der Dokumentation:

- V1.0 im November 2014
- V1.1

Inhaltsverzeichnis

Einleitung	4
Grundsätzlicher Aufbau.....	4
Funktionsbeschreibung.....	5
Das MoodLight	6
Konstante Farbe	6
Farbwechsel der „7 Grundfarben“	6
Farbwechsel der DIY (Do It Yourself) Farben	6
Farbwechsel im „Multi Sinus“ Mode	7
Bedienung und Parametrierung im AmbiLight Modus	8
Parametrierung des PitschuLights per Terminal	8
Beschreibung der Parameter	9
Tipps zur richtigen Einstellung der Parameter	12
Parametrierung über die IR-Bedienung	12
Die Hardware	14
Funktionsweise des AmbiLights.....	15

Einleitung

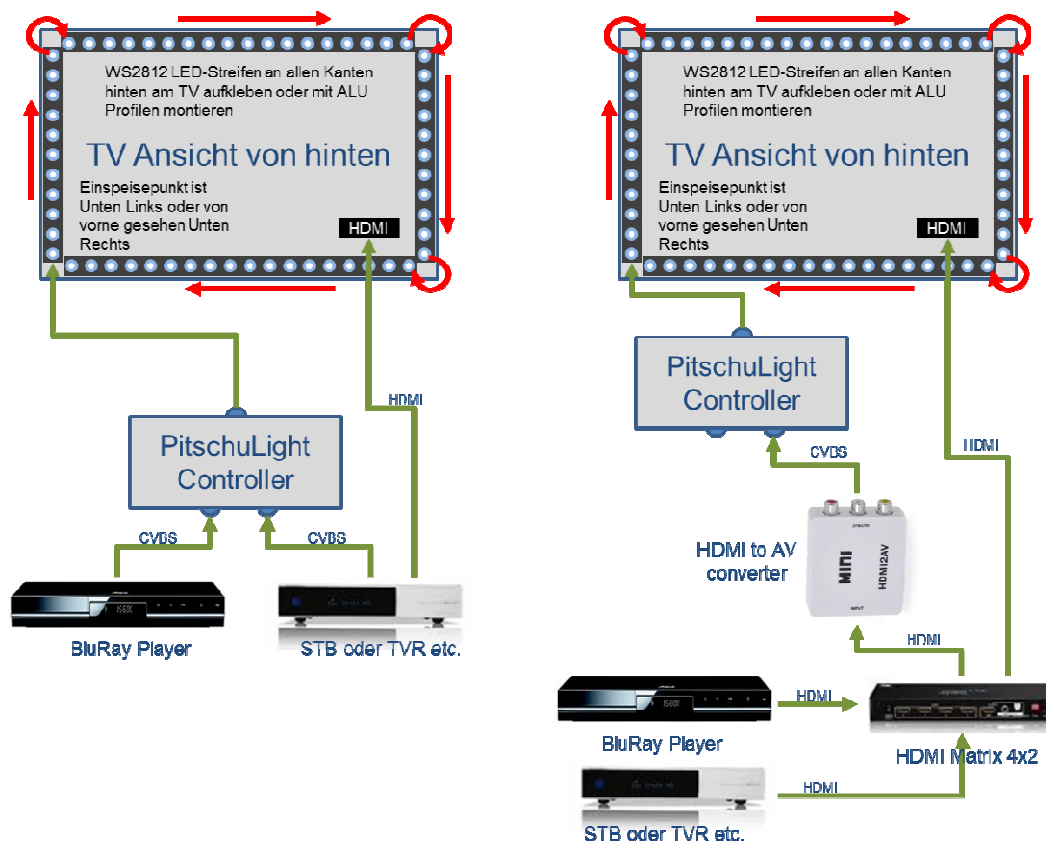
Angefangen habe ich Mitte 2013, als ich meinen Philips Fernseher gegen einen neuen Samsung HD-TV getauscht habe. Da der Philips ein eingebautes AmbiLight hatte (allerdings nur mit sehr grober links /rechts Unterscheidung), wurde mir schnell klar, dass ich wieder ein AmbiLight haben möchte. Der Einfluss eines solchen dynamischen Backlights auf das Seherlebnis ist erstaunlich.

Auf dem Markt habe ich keine fertige Lösung gefunden und die Nachbaulösungen basieren fast alle auf irgendwelchen PC/Linux STB Addons oder aber entsprechende Infos (Schaltpläne und Software) sind nicht frei verfügbar. Ich wollte auf jeden Fall eine Lösung, die mit minimaler Hardware auskommt, selbständig funktioniert und sich automatisch mit dem Fernseher/STB ein/ausschaltet.

Da ich noch ein STM32F4discovery Board rumliegen und bereits einige Spielereien damit unternommen hatte, kam mir die Idee, auf Basis des enthaltenen DCMI-Interfaces evtl. eine Lösung finden zu können. Einige Recherchen brachten mich dann zum Video-Prozessor TVP5150AM1 von Texas Instruments. Als Gespann sind sie ideal für die Umsetzung eines AmbiLight geeignet. Da ich vor dem Fernseher noch einen TV-Recorder betriebe, der auch einen Chinch-Ausgang besitzt, war die Notwendigkeit von HDMI-Verarbeitung nicht gegeben. Inzwischen sind auch verschiedene Varianten von HDMI-Splittern/Matrizen sowie CVBS-Konvertern getestet worden, so dass eine native HDMI-Unterstützung gar nicht erforderlich ist.

Grundsätzlicher Aufbau

Die folgenden Bilder zeigen Möglichkeiten des grundsätzlichen Aufbaus. Die linke Variante kommt zum Einsatz, wenn die angeschlossenen Geräte bereits einen PAL tauglichen Ausgang (CVBS) haben. Das rechte Bild zeigt eine Möglichkeit, mit Hilfe einer HDMI-Matrix und eines HDMI2AV Konverters ein solches PAL-Signal zu erzeugen und gleichzeitig den Bildschirm mit HDMI zu versorgen.



Die Stromversorgungen für PitschuLight und LED-Streifen sind hier nicht gezeigt.

Unabhängig von der gewählten Variante hat der PitschuLight-Controller die Aufgabe, Videodaten zu analysieren und in einen Datenstrom umzuwandeln, der für einen WS2812 LED-Streifen geeignet ist. Der LED-Streifen wird dabei ringförmig auf die Rückseite des TV geklebt oder z.B. mit entsprechenden ALU-Profilen, die das Licht schön verteilen, montiert. Der Anfang des Streifens ist von hinten gesehen Unten/Links, geht dann nach oben, knickt dann nach rechts ab usw. bis er wieder unten links ankommt. In den Ecken wird er einfach durchtrennt und die 3 Adern des Streifens mit kurzen Kabeln um die Ecke zum nächsten Stück verlegt. Die Länge des LED-Streifens ist im PL einstellbar zwischen 4 LED und 312 LEDs (312 = 2x60 Seitenkanten plus 2x96 oben und unten). Den unteren Streifen kann man auch weglassen, z.B. wenn das Kabelzeugs etc. was hinter dem Fernseher steht nicht beleuchtet werden soll.

Um die richtige Anzahl von LEDs zu bestimmen, sollte man die Glasscheibe des TV ausmessen und entsprechend lange Stücke von WS2812-Streifen abschneiden. Evtl. müssen die etwas kürzer sein, da die Montagemöglichkeiten auf der Rückseite dies erfordern. Kleinere Abweichungen zwischen Abmessungen des Schirms und Länge der Streifen ist nicht tragisch, bei größeren Abweichungen kann man aber evtl. deutliche Verschiebungen zwischen Bildinhalt und LEDs schon sehen.

Funktionsbeschreibung

Das PitschuLight lässt sich in zwei verschiedenen Modi betreiben:

1. im AmbiLight Modus werden die LEDs anhand der Bildinformation eines eingespeisten Videosignals angesteuert. Dazu wird das Videobild analysiert und sowohl Farben als auch Helligkeit des äußeren Bildrandes auf die entsprechenden LEDs projiziert. So wird optisch das Fernsehbild nach allen Seiten hin in den Raum erweitert.
2. Im MoodLight Modus werden die LEDs unabhängig von einem Eingangssignal so angesteuert, dass sich entweder eine konstante Farbe oder auch zeitlich wechselnde Farben einstellen. In weiteren Varianten lassen sich auch verschiedene, zeitlich wechselnde Muster darstellen.

Das PitschuLight wird über den roten ON/OFF Button der Fernbedienung oder alternativ über den blauen Knopf auf dem STM32F4discovery Board ein EIN und AUS geschaltet. Dazu ist einer der genannten Button kurz zu betätigen. Im ausgeschalteten Zustand werden lediglich die LEDs nicht mehr angesteuert und bleiben dunkel. Weiterhin aktiv bleibt jedoch die automatische Kanalsuche des AmbiLights (sofern über den Parameter V = Videosource aktiviert). In diesem Fall zeigen die LED in der linken bzw. rechten unteren Ecke an, auf welchem Kanal gerade nach einem Videosignal gesucht wird. Ist auf keinem der Kanäle ein Videosignal vorhanden, so wechseln die beiden LEDs im 2-Sekundenrhythmus. Im eingeschalteten Zustand werden die LEDs gemäß dem aktiven Modus (AmbiLight oder MoodLight) entsprechend angesteuert.

Die Umschaltung zwischen AmbiLight und MoodLight Modus erfolgt durch die Betätigung des roten ON/OFF Buttons für länger als 2 Sekunden. Der Modus bleibt solange aktiv, bis wieder der ON/OFF Button für mehr als 2 Sekunden betätigt wird.

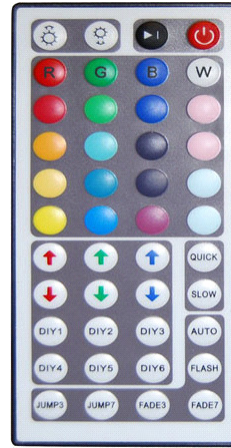
Die auf dem STM32F4discovery Board befindlichen 4 LEDs (rot, blau, grün und orange) dienen als Funktionskontrolle für diverse AmbiLight-Operationen. Sie zeigen folgende Zustände an:

- Die grüne LED zeigt Empfangsaktivität auf dem IR-Empfänger an, d.h. sie sollte flackern, wenn auf der IR-Bedienung eine Taste gedrückt wird.
- Die rote LED zeigt an, ob das horizontale Synchronsignal (HSYNC) vom Videoprozessor aktiv ist.
- Die orange LED zeigt an ob das vertikale Synchronsignal (VSYNC) vom Videoprozessor aktiv ist.
- Die blaue LED wird zu verschiedenen Zeitpunkten innerhalb der Hauptschleife des Programms ein und aus geschaltet. Sie kann z.B. für Zeitmessungen mit Hilfe eines Oszilloskops verwendet werden.

Das MoodLight

Alternativ zum AmbiLight Modus kann das PitschuLight als MoodLight (= Stimmungslicht) verwendet werden. Das folgende Bild zeigt die Belegung der FB-Tasten im MoodLight-Modus und dient zum besseren Verständnis der danach folgenden Beschreibung des MoodLights:

Quick Reference Guide for pitschu's AmbiLight V1.2		
Key	Moodlight	
	Normal modes	Multi Sinus Mode
On/Off	>2 sek = Standby; Toggle Ambi/Mood	
Mode Black	--	
Brightness UP Key	increase brightness	
Brightness DOWN key	Decrease brightness	
Red [R]	Set fixed color	
Green [G]	Set fixed color	Set sinus period
Blue [B]	Set fixed color	Set Amplitude
White [W]	Set fixed color	Set base brightness
Other color keys	Set fixed color	
Red UP	RED color++	increase RED value
Red DOWN	RED color--	Decrease RED value
Green UP	GREEN color++	increase GREEN value
Green DOWN	GREEN color--	Decrease GREEN value
Blue UP	BLUE color++	increase BLUE value
Blue DOWN	BLUE color--	Decrease BLUE value
Quick key	increase fading / sinus speed	
Slow key	Decrease fading / sinus speed	
DIY 1	Select / Save (>2sek) current color / Setting	
DIY 2		
DIY 3		
DIY 4		
DIY 5		
DIY 6		
Auto key	Set actual value to default	
Flash key		
Jump3 key	Set mode "Multi sinus waves"	
Jump7 key	---	
Fade3 key	Set mode "Fade DIY colors" up to 6 colors in DIY	
Fade7 key	Set mode "Fade 7 base colors"	



Zur Aktivierung des MoodLights wird die ON/OFF Taste für länger als 2 Sekunden betätigt, um vom AmbiLight in den MoodLight Modus zu wechseln.

Die Grundhelligkeit des MoodLights wird über die beiden Helligkeitstasten (oben links auf der FB) eingestellt. Diese Tasten gelten unabhängig vom gewählten Modus.

Das MoodLight kann grundsätzlich in folgenden Modi betrieben werden:

Konstante Farbe

Hierzu wird eine der Farbtasten gedrückt, wonach diese Farbe nach der eingestellten Fading-Zeit angezeigt wird. Die Fading-Zeit kann über die Tasten QUICK/SLOW verkürzt (= schneller Farbwechsel) oder verlängert (= langsamer Farbwechsel, bis zu 30 Sekunden) werden. Über die roten, grünen und blauen Pfeiltasten kann der jeweilige RGB Wert der angezeigten Farbe verändert und somit eine beliebige Farbe eingestellt werden. Durch längeres Drücken (> 2 Sekunden) einer der 6 DIY (Do It Yourself) Tasten wird die aktuell angezeigte Farbe auf einen der 6 verfügbaren Speicherplätze gespeichert werden. Kurzes Drücken einer DIY Taste stellt dann die gespeicherte Farbe dar.

Durch sehr langes Drücken (> 4 Sekunden) einer DIY Taste wird die darunter abgespeicherte Farbe wieder gelöscht und der DIY Speicher geleert.

Farbwechsel der „7 Grundfarben“

Über die Taste FADE7 werden die „7 Grundfarben“ in der durch die Tasten QUICK/SLOW festlegbaren Fading-Zeit durchlaufen.

Farbwechsel der DIY (Do It Yourself) Farben

Über die Taste FADE3 werden die unter den DIY Tasten abgespeicherten Farben in der durch die Tasten QUICK/SLOW festlegbaren Fading-Zeit durchlaufen. Die Fading-Zeit kann über die Tasten QUICK/SLOW verkürzt (= schneller Farbwechsel) oder verlängert (= langsamer Farbwechsel, bis zu 30 Sekunden) werden.

Im Abschnitt „Konstante Farbe“ (s.o.) ist die Festlegung der DIY Farben beschrieben. Wenn nicht alle DIY Speicher mit einer Farbe belegt sind, erfolgt der Farbwechsel über entsprechend weniger DIY Farben.

Farbwechsel im „Multi Sinus“ Mode

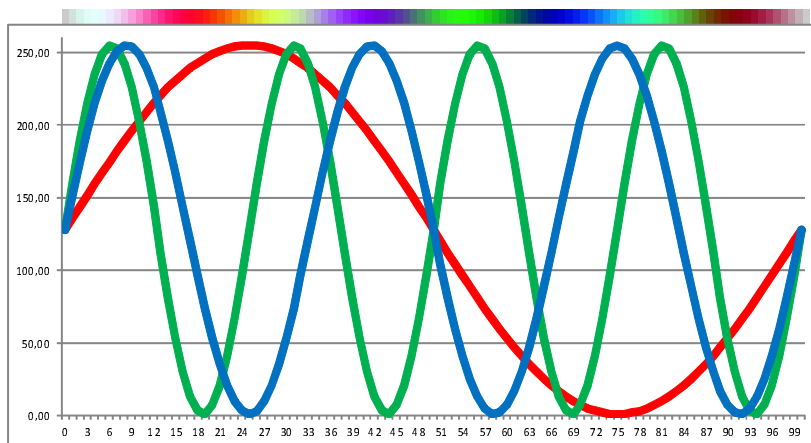
Über die Taste JUMP3 wird der sog. Multi-Sinus Modus gestartet. Dieser Modus bietet eine praktisch unbegrenzte Vielzahl zur Darstellung verschiedener Muster, bewegte Effektlichter oder Laufeffekte. Im Grunde werden dabei die 3 Grundfarben der LEDs über konfigurierbare Sinusschwingen angesteuert, wobei neben der Periodenzeit auch die Amplitude und Grundhelligkeit eingestellt werden kann. Im Folgenden sollen die Zusammenhänge kurz beschrieben werden.

Angenommen, der LED-Streifen besteht aus 100 WS2812 LEDs. Jede dieser LED hat eine rote, eine grüne und eine blaue Teil-LED. Im Multi-Sinus Modus werden nun die 100 roten Teil-LEDs mit einer sinusförmigen Helligkeitskurve angesteuert, bei welcher sowohl die Frequenz, die Amplitude und die Grundhelligkeit einstellbar sind. Getrennt von der „roten“ Kurve gibt es entsprechende Kurven für die 100 grünen und die 100 blauen Teil-LEDs. Die folgende Tabelle zeigt diese 9 Parameter in der Übersicht mit ihren Standardwerten.

	Taste	ROT	GRÜN	BLAU
Frequenz	G	1	4	3
Amplitude	B	127	127	127
Helligkeit	W	128	128	128

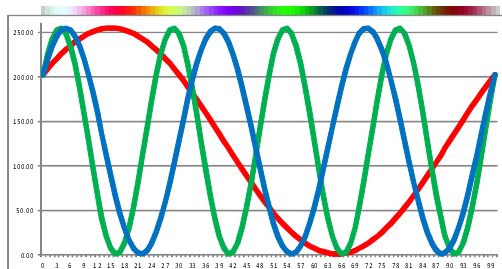
Die Änderung der Werte erfolgt über die entsprechend farbig markierten Pfeiltasten. Um z.B. die Amplitudenwerte zu verändern, wird zunächst die blaue Farbtaste mit dem aufgedruckten B gedrückt. Anschließend lassen sich über die 6 Pfeiltasten die Werte der jeweiligen Farbe aufwärts oder abwärts verstellen. Eine grüne LED an der Oberkante des LED-Streifens zeigt den aktuellen Wert als Prozentwert an (links = niedrig, rechts = hoch).

Das folgende Bild zeigt schematisch die den Tabellenwerten entsprechende Ansteuerung der LEDs:



Am oberen Rand des Bildes ist die dann sichtbare, zusammen-gemischte Farbe der LEDs dargestellt. Für jede der 100 LEDs sind die jeweiligen Rot / Grün / Blau Werte der Teil-LEDs abgetragen. Die technisch möglichen Werte jeder LED-Farbe liegen zwischen 0 (= dunkel) bis 255 (= volle Helligkeit). Die Grundhelligkeit von 128 liegt also bei der halben

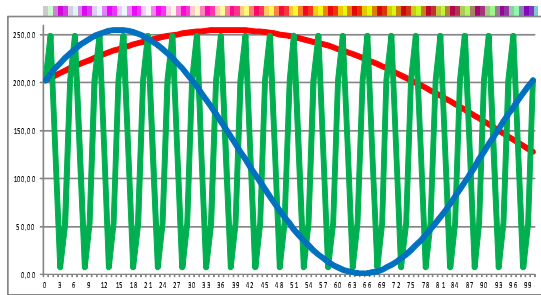
Maximalaussteuerung (= Mitte der Grafik). Die Frequenz von ROT ist 1 Schwingung über alle 100 LEDs, GRÜN entsprechend 4 Schwingungen und BLAU 3 Schwingungen. Die Anzahl der Schwingungen kann von ca. 0,2 bis zu 44 eingestellt werden.



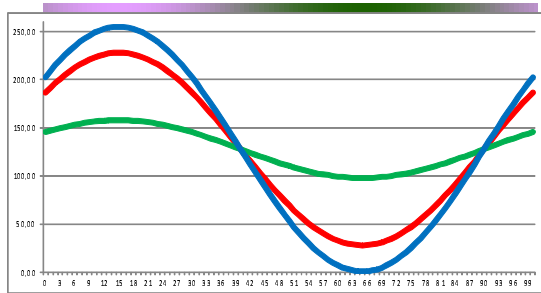
Über die QUICK/SLOW Tasten wird des Weiteren die Durchlaufzeit für eine sich ständig ändernde Phase eingestellt. Dabei durchläuft die Phase einmal das komplette Bild innerhalb der eingestellten Durchlaufzeit. Sie kann von 0,25 Sekunden bis zu 30 Sekunden betragen. Das nebenstehende Bild zeigt die Darstellung wie oben, jedoch mit veränderter Phase. Die Kurven beginnen nun nicht mehr bei der ersten LED, sondern sind verschoben. Durch die sich ständig ändernde Phase entsteht so ein schöner Bewegungseffekt, der das Leuchtmuster entlang der LEDs

verschiebt.

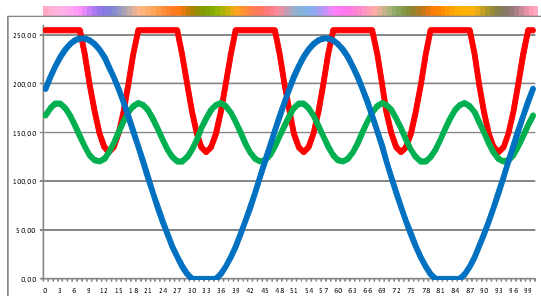
Durch beliebige Veränderung der genannten Parameter lassen sich völlig unterschiedliche Beleuchtungsmuster erzeugen. Die folgenden Bilder zeigen einige Beispiele:



	Taste	ROT	GRÜN	BLAU
Frequenz	G	0,4	20	1
Amplitude	B	127	127	127
Helligkeit	W	128	128	128



	Taste	ROT	GRÜN	BLAU
Frequenz	G	1	1	1
Amplitude	B	100	30	127
Helligkeit	W	128	128	128



	Taste	ROT	GRÜN	BLAU
Frequenz	G	5	6	2
Amplitude	B	100	30	127
Helligkeit	W	230	150	120

Eine einmal eingestellte Kombination von Werten kann durch langes Drücken (> 2 Sekunden) einer der DIY Tasten zugeordnet und abgespeichert werden. Durch kurzes Drücken der DIY Taste wird die Parameterkombination geladen und zur Darstellung verwendet.

Bedienung und Parametrierung im AmbiLight Modus

Die Bedienung des PitschuLights im AmbiLight Modus erfolgt entweder über die Terminalverbindung oder über die IR-Fernbedienung.

Bei Betrieb als MoodLight erfolgt die Bedienung ausschließlich über die IR-Bedienung.

Parametrierung des PitschuLights per Terminal

Die Parametrierung des PitschuLights erfolgt am besten über eine Terminalverbindung, da in diesem Fall die genauen Einstellungen der veränderten Variablen am Terminal sichtbar sind. Eine Parametrierung per IR-Bedienung ist zwar möglich, aber ein Feed-Back über die neuen Einstellungen erfolgen lediglich über eine grün leuchtende LED an der Oberkante des Fernsehers.

Die folgende Auflistung zeigt die unterstützten Kommandos im Terminalmodus:

```
Usage: use +/- keys to set val; d=default
F=Hue, S=Saturation, B=Brightness, C=Contrast
L=Left, W=Width, T=Top, H=Height
I=I-factor of integrator (128 = MAX)
E=# of slots aggregated for LED strip (1..10)
```



```

G=Frame count for dynamic 'black border' detection (0=OFF; 1..200)
X=virtual image width in blocks
Y=virtual image height in blocks
P=Physical image width: # of LEDs
R=Physical image height: # of LEDs
V=select video source (1 or 2)
Q=show info about Dyn Matrix
N=restart TVP5150 and show reg info
A=set TVP5150 auto gain control ON/OFF
M=set frame delay time (0-20 frames)
0,1 or 2: Set input channel 1 or 2; 0 = Auto

```

Grundsätzlich wird zunächst mit einem Buchstaben (Groß oder Klein) ein zu parametrierender Wert festgelegt. Anschließend wird dieser Wert durch wiederholtes Drücken der ‚+‘ oder ‚-‘, Taste um eins erhöht bzw. um eins erniedrigt. Wird statt ‚+‘ oder ‚-‘, der Buchstabe ‚d‘ betätigt, so wird der Parameter auf seinen Defaultwert gesetzt. Nach jedem Bestätigen einer dieser Tasten wird der neue Wert des Parameters angezeigt.

Ca. 8 Sekunden nach der letzten Änderung eines Parameters wird der aktuelle Stand der Parameter in den FLASH-Speicher des STM32F407 geschrieben, so dass die Parameter auch nach Abschalten der Versorgungsspannung beim späteren Einschalten wieder zur Verfügung stehen und verwendet werden. Nach Änderung eines Parameters sollte daher nicht unmittelbar danach die Versorgung unterbrochen werden, damit der Speichervorgang noch durchgeführt werden kann.

Beschreibung der Parameter

Im Folgenden werden alle Einstellparameter und ihre Auswirkungen im PitschuLight Modus kurz beschrieben. Im Anschluss an die Kurzbeschreibung sind die MIN, MAX und Defaultwerte der Parameter angegeben.

F=Hue, S=Saturation, B=Brightness, C=Contrast

Diese 4 Parameter spiegeln die gleichlautenden Registerwerte des TVP5150 wider. Die eingestellten Werte werden direkt in die Register übernommen. Einzelheiten zu diesen Parametern kann man dem Datenblatt entnehmen, ansonsten sollten sie klar sein, da es auch z.B. im Setup des Fernsehers üblicherweise diese 4 Parameter gibt.

Para	MIN	MAX	DEF
F	-127	+125	0
S	1	255	128
B	1	255	60
C	1	197	80

P=Physical image width: # of LEDs

R=Physical image height: # of LEDs

Diese Parameter legen fest, wie viele LEDs physikalisch angeschlossen sind. Sie müssen exakt mit der Anzahl der LEDs an den horizontalen bzw. vertikalen LED-Streifen entsprechen.

Para	MIN	MAX	DEF
P	1	96	48
R	1	60	28

L=Left, W=Width, T=Top, H=Height

Diese 4 Parameter legen fest, welcher Bildausschnitt des Fernsehers mit den LEDs des PitschuLights abgebildet werden sollen. Idealerweise sind diese Werte so einzustellen, dass z.B. ein Fernsehbild, welches in allen 4 Ecken einen begrenzten, farbigen Bereich zeigt, genau so auf den LED Streifen erscheint, d.h. die 4 Ecken des LED-Rings zeigen genau die gleichen Farben wie der Fernseher. Sind die 4 Parameter falsch eingestellt, so ergeben sich z.B. Verschiebungen der LEDs oder die LEDs geben die Farbbereiche gar nicht wieder.

Technisch gesehen definieren diese Parameter das sog. Crop-Window (s.a. Funktionsweise), also den Ausschnitt des vorgegebenen PAL-Windows, welcher dargestellt werden soll. Eine Bildzeile des PAL-Signals hat 864 Pixel, von denen jedoch nur 720 sichtbare Inhalte haben.

Ein konkreter Fernseher schneidet hiervon i.d.R. noch etwas ab, so dass der wirklich sichtbare Bereich z.B. bei Pixel 80 beginnt und nur 700 Pixel breit ist. Diese beiden Werte sind entsprechend einzustellen. Für die vertikale Einstellung gilt Analoges. Für Tipps zur Einstellung siehe entsprechendes Kapitel unten.

Para	MIN	MAX	DEF
L	40	200	80
W	200	740	696
T	4	150	16
H	40	312	288

I=I-factor of integrator (128 = MAX)

Über diesen Parameter kann die Anpassungsgeschwindigkeit des PitschuLights an den Bildinhalt eingestellt werden. Bei großen Werten ist die Anpassung sehr schnell, bei kleinen Werten sehr langsam. So reagiert das PitschuLight bei einer Einstellung von 128 unmittelbar auf Bildänderungen, was bei schnellen Bildwechseln sehr störend wirken kann. Bei kleinen Werten ergibt sich ein gemächliches Anpassen des PitschuLights an den Bildinhalt, was als zu schleppend empfunden werden kann. Die Standardeinstellung von 32 ist ein guter Kompromiss.

Para	MIN	MAX	DEF
I	1	128	32

E=# of slots aggregated for LED strip (1..11)

Das PitschuLight versteht sich als Erweiterung des Bildes nach außen und nimmt dazu die äußeren Bildinhalte als Vorgabe. Dies kann z.B. nur der äußerste Rand sein (Einstellung = 1) oder ein breiterer äußerer Rand dessen Bildinhalte als Mischwerte an das PitschuLight gegeben werden. Technisch definiert dieser Parameter die Anzahl der virtuellen Blöcke, die gemischt und an das PitschuLight weitergegeben werden (s.a. Funktionsbeschreibung).

Para	MIN	MAX	DEF
E	1	11	4

G=Frame count for dynamic 'black border' detection (0=OFF; 1..200)

Das PitschuLight passt sich permanent an den sichtbaren Bildbereich auf dem Fernseher an. Dieser Bildbereich kann durch unterschiedliche Formatanpassungen und –skalierungen bei verschiedenen TV-Inhalten sehr unterschiedlich sein. Das vom PitschuLight verwendete PAL-Format hat ein Bildformat von 4:3. Heute übliche Fernseher verwenden i.d.R. das Format 16:9 und Bildinhalte wie Kinofilme verwenden z.B. das Format 5:3. Zur Darstellung der Inhalte werden die Formate entweder in der Höhe oder in der Breits skaliert oder aber es wird nur ein Bildausschnitt wirklich sichtbar. Je nach verwendeter Skalierung kann es auf dem Bildschirm zu schwarzen Bereichen links und rechts oder oben und unten kommen. Da das PitschuLight den äußeren Bildbereich zur Ansteuerung der LEDs verwendet, würden z.B. die LEDs an der Oberkante immer dunkel bleiben, wenn ein Kinofilm mit 5:3 Format auf dem Bildschirm mit schwarzen Balken oben und unten gezeigt wird. Um das zu vermeiden sucht das PitschuLight permanent nach schwarzen Bereichen an allen 4 Seiten und passt so dynamisch die 4 Seiten an den „beleuchteten“ Bildbereich an.

Über den Parameter „G“ kann man die Anpassungsgeschwindigkeit der Suchfunktionen beeinflussen. Bei G=0 ist der Mechanismus abgeschaltet so dass keine Anpassung stattfindet. Dieser Wert wird eigentlich nur benutzt, während andere Parameter (L, W, T und H) eingestellt werden. Werte Größer 0 geben die Anzahl der Frames (Übertragene Bilder) an, deren Inhalte gemittelt werden und zur Bestimmung der Balken herangezogen werden. Ein Wert von 200 bedeutet also, dass über 200 Frames (ca. 8 Sekunden) in die Berechnung eingehen. Mit großen Werten wird z.B. verhindert, dass eine dunkle Filmszene, deren Bildränder sowieso fast schwarz sind, nicht dazu führt, dass das PitschuLight sich sofort darauf einstellt und möglicherweise einen mehr in der Bildmitte liegenden Bildteil in den LEDs wiedergibt. Dies würde den Gesamteindruck stark stören.

Durch die Software wird der Suchbereich nach schwarzen Balken auf ca. ¼ der Bildhöhe und ca. 1/6 der Bildbreite begrenzt.

Para	MIN	MAX	DEF
G	0	200	100

X=virtual image width in blocks

Y=virtual image height in blocks

Das PitschuLight arbeitet intern mit einem groben Abbild des Bildschirms. Dieses virtuelle Bild wird verwendet um die einzelnen Berechnungsschritte (s.a. Funktionsweise) vom Videosignal bis zur LED Ansteuerung durchzuführen. Letztendlich definieren diese beiden Parameter die Auflösung des virtuellen Bildes und damit auch die Auflösung der LEDs. Die maximalen Werte der Parameter sind X = 64 und Y = 40. Das bedeutet, dass z.B. die LEDs an der Ober- und Unterkante maximal 64 verschiedene Bereiche darstellen können, auch wenn mehr LEDs verbaut sind. Stellt man dagegen z.B. die Werte X=2 und Y=2 ein, so unterscheidet das PitschuLight nur noch jeweils 2 Bereiche in X und Y Richtung. Man sieht dann z.B., dass die linke Hälfte der LEDs an der Oberkante alle gleich leuchten und den Farbmittelwert der linken Bildhälfte wiedergeben. Entsprechendes gilt für die rechte Hälfte.

Para	MIN	MAX	DEF
X	1	64	64
Y	1	40	40

V=select video source (1 or 2)

0,1 or 2: Set input channel 1 or 2; 0 = Auto

Nach Anwahl des Parameters „V“ kann über die Tasten „1“ und „2“ die entsprechende Eingangsbuchse als Signalquelle ausgewählt werden. Über die Auswahl „0“ wird das PitschuLight in den Scan-Modus versetzt. Hierbei sucht das PitschuLight permanent auf beiden Eingängen nach einem Signal. Sobald ein Signal gefunden ist, wird das Signal zur LED-Ansteuerung verwendet. Fällt das Signal aus (Quelle wird z.B. abgeschaltet), so geht das PitschuLight wieder in den Scan-Modus.

Zwischen den Eingängen wird nicht unterschieden, d.h. es gibt keine Vorrangschaltung für einen der Kanäle (First come first serve).

Q=show info about Dyn Matrix

Dies ist kein Parameter, sondern lediglich eine Debug-Funktion des PitschuLights. Es werden die Inhalte der Suchvariablen der dynamischen „Schwarzbalken-Anpassung“ auf dem Terminal ausgegeben.

N=restart TVP5150 and show reg info

Wie der Name sagt, wird der Videoprozessor TVP5150 auf die RESET-Werte zurückgesetzt und neu initialisiert. Anschließend werden alle Register des TVP5150 am Terminal ausgegeben.

A=set TVP5150 auto gain control ON/OFF

Der Videoprozessor TVP5150 hat eine eingebaute Funktion zur automatischen Anpassung an das Bildsignal. Diese sorgt z.B. dafür, dass ein schwaches Bildsignal ausreichend verstärkt wird, um das PitschuLight sauber zu bedienen. Andererseits kann sie aber auch dazu führen, dass dunkle Bildbereiche in den LEDs zu hell erscheinen.

Über diesen Parameter lässt sich die genannte Anpassungsfunktion EIN und AUS schalten.

M=set frame delay time (0-20 frames)

Moderne Fernseher bereiten die eingehenden Videosignale über mehr oder weniger komplexe Bildprozessoren auf und zeigen sie an. Die Zeit zur Aufbereitung vom Signal zum LCD/Plasma benötigt einige Zeit, d.h., wenn das Signal einen plötzlichen Helligkeits- oder Farbwechsel beinhaltet, ist das erst nach diversen Millisekunden am Schirm sichtbar. Das PitschuLight ist in der Lage, diese Farb- und Helligkeitswechsel quasi in Echtzeit ohne Verzögerungen an den LEDs darzustellen. Um solche Wechsel zu synchronisieren, also an den LEDs und dem Bildschirm gleichzeitig erscheinen zu lassen, kann über diesen Parameter eine Verzögerungszeit im PitschuLight eingestellt werden. Diese Verzögerungszeit kann zwischen 0 (= keine Verzögerung) und bis zu 20 Frames (ca. 0,8 Sekunden) betragen. Der Parameter sollte so eingestellt sein, dass kein Zeitunterschied zwischen TV-Bildschirm und PitschuLight mehr sichtbar ist.

Para	MIN	MAX	DEF
------	-----	-----	-----

Tipps zur richtigen Einstellung der Parameter

Im Folgenden werden alle relevanten Einstellparameter und ihre Auswirkungen im AmbiLight Modus kurz beschrieben (für Beschreibung der Parameter s. o.).

1. Erst die Anzahl der vorhandenen LEDs mit P und R einstellen.
2. Die Parameter X und Y sollten die gleichen Werte wie P und R haben. Größere Werte haben keinen Mehrwert; kleinere Werte machen die LED-Darstellung gröber (mehrere LEDs mit gleicher Farbe).
3. Zum Einstellen der Links/Rechts bzw. Oben/Unten Grenzen (L/W/T/H) sollte die dynamische Balkenerkennung abgeschaltet sein, daher den Parameter G auf 0 setzen.
4. Der Parameter E sollte 1 sein. Damit zeigen die LEDs nur den jeweils äußersten Streifen des Bildes an.
5. Die Randeinstellung (Parameter L, W, T und H) am besten mit einem Standbild (ideal sind Comics) vornehmen. Die Grenzen so einstellen, dass die LEDs die Farben des äußeren sichtbaren Bildrandes anzeigen. Das Bild muss Vollbild, darf also keine Balken haben. Folgendes Vorgehen wird empfohlen:
 - L sehr klein einstellen und dann den linken Bildrand suchen, indem L vergrößert wird. Dabei muss man auch auf die linken LEDs am oberen und unteren Streifen achten. Wenn LEDs fehlen oder schwach flackern, dann ist der Wert noch zu klein. Danach den Parameter W sehr groß einstellen und den rechten Rand suchen, indem W verkleinert wird. Auch hier muss man auf die rechten LEDs oben und unten achten. Fehlen LEDs oder flackern nur schwach, dann ist W noch zu groß.
 - Für die Oben/Unten Einstellung analog dazu T sehr klein einstellen und dann den oberen Rand suchen, indem T vergrößert wird. Dabei achtet man auch auf die oberen LEDs rechts und links. Anschließend H sehr groß einstellen und den unteren Rand suchen, indem H langsam wieder verkleinert wird. Dabei achtet man auch auf die unteren LEDs links und rechts.
6. Nach Einstellung der Grenzen sollten die Parameter G und E wieder zurückgestellt werden (siehe Parameterbeschreibung).

Parametrierung über die IR-Bedienung

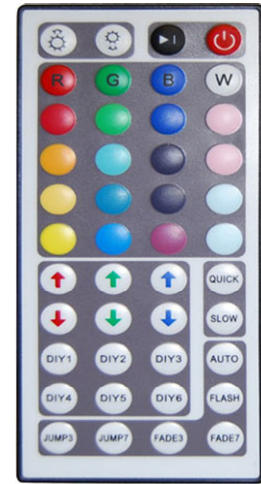
Über die IR-Fernbedienung können die meisten der PitschuLight Parameter für den AmbiLight Modus eingestellt werden (Beschreibung der Parameter siehe vorheriges Kapitel).

Im MoodLight Modus erfolgt die Bedienung grundsätzlich über die IR-Fernbedienung. Zur Beschreibung der unterschiedlichen Lichtvarianten siehe Kapitel „Moodlight“.

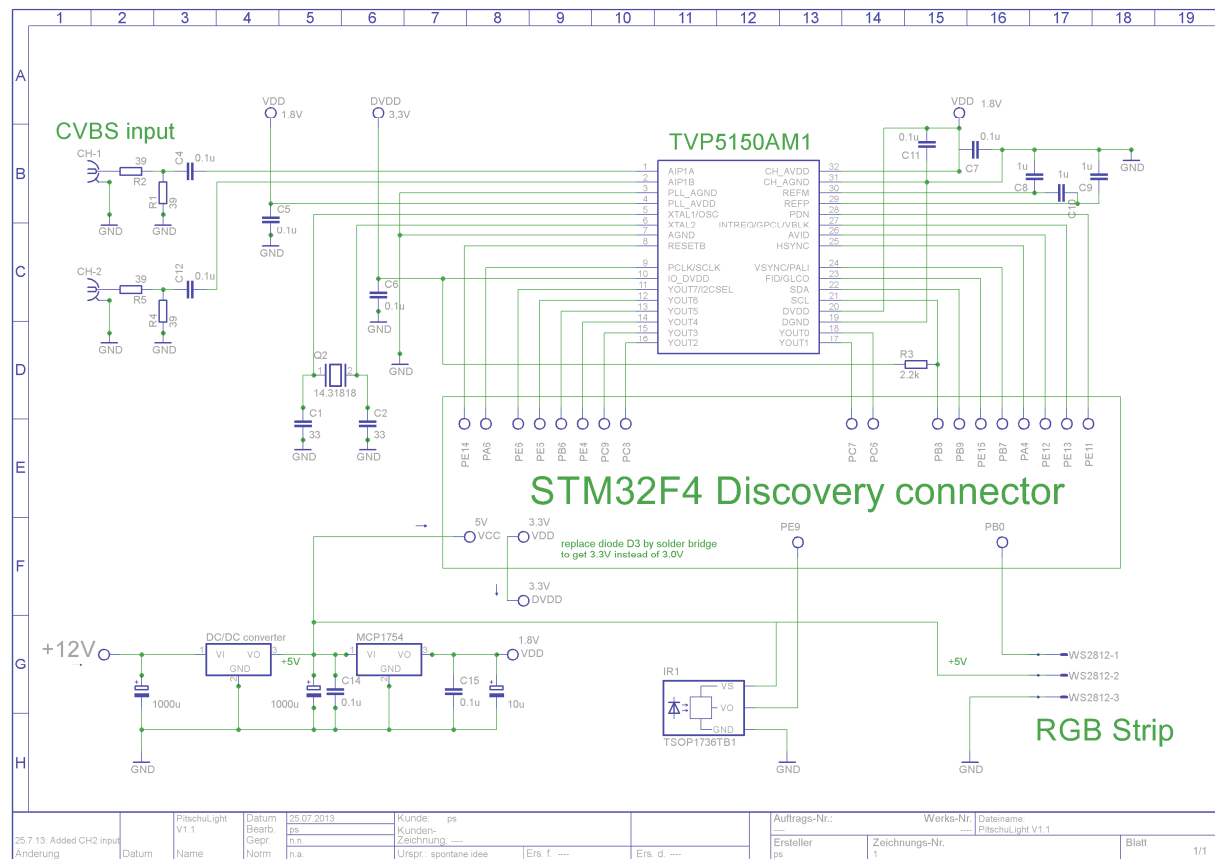
Da bei der IR-Bedienung keinerlei Ausgabe der aktuellen Parameterwerte erfolgen kann (wie im Terminal Modus), wird stattdessen die LED Leiste der Oberkante als Anzeige benutzt. Dabei leuchtet eine LED in grüner Farbe mit maximaler Helligkeit. Die ganz linke LED zeigt einen Wert von 0%, die ganz rechte den Wert 100% an. Die Prozentwerte beziehen sich auf den jeweiligen Einstellbereich des aktuell bearbeiteten Parameterwertes.

Die folgende Tabelle enthält die Zuordnungen aller Tasten der verwendeten 44-Key IR-Fernbedienung in den beiden Betriebsmodi des PitschuLight.

Key	Ambilight	Moodlight	
		Normal modes	Multi Sinus Mode
On/Off	>2 sek = Standby; Toggle Ambi/Mood	>2 sek = Standby; Toggle Ambi/Mood	
Mode Black	Video select: Chan 1 / 2 / Auto	--	
Brightness UP Key	Increase actual value	Increase brightness	
Brightness DOWN key	Decrease actual value	Decrease brightness	
Red [R]	Hue control -127..125 [0]	Set fixed color	
Green [G]	Saturation 1..255 [128]	Set fixed color	Set sinus period
Blue [B]	Contrast 1..197 [80]	Set fixed color	Set Amplitude
White [W]	Brightness 1..255 [60]	Set fixed color	Set base brightness
Other color keys	---	Set fixed color	
Red UP	Set left border 10..100 [84]	RED color++	Increase RED value
Red DOWN	Set image width [668]; max = 840	RED color--	Decrease RED value
Green UP	Set top border 4..100 [16]	GREEN color++	Increase GREEN value
Green DOWN	Set image high [288]; max = 306	GREEN color--	Decrease GREEN value
Blue UP	Set image blocks X 1..64 [64]	BLUE color++	Increase BLUE value
Blue DOWN	Set image blocks Y 1..40 [40]	BLUE color--	Decrease BLUE value
Quick key	Lower integration time 1..128 [32]	Increase fading / sinus speed	
Slow key	Increase integration time 1..128 [32]	Decrease fading / sinus speed	
DIY 1	---	Select / Save (>2sek) current color / Setting	
DIY 2	---		
DIY 3	---		
DIY 4	---		
DIY 5	---		
DIY 6	---		
Auto key	Set actual value to default	Set actual value to default	
Flash key	Set border width in blocks 1..11 [4]		
Jump3 key	Toggle AGC (auto gain control) on/off	Set mode "Multi sinus waves"	
Jump 7 key	---	---	---
Fade3 key	---	Set mode "Fade DIY colors" up to 6 colors in DIY	
Fade7 key	Set delay time in frames 0..20 [0]	Set mode "Fade 7 base colors"	



Die Hardware



Das Bild zeigt nur die zusätzlich zum STM32F4discovery Board benötigte Schaltung. Das STM Board besitzt 2 Pfostenstecker, auf die eine Zusatzplatine mit den gezeigten Bauteilen aufgesteckt wird. Die Beschaltung des TVP5150 entspricht der im Datenblatt angegebenen Standardbeschaltung. Über 2 Chinchbuchsen können CVBS-Signale (auch FBAS genannt) z.B. vom Fernseher, TV-Recorder und BluRay-Spieler eingespielt werden. Eine automatische oder manuelle Kanalschaltung des TVP5150 ermöglicht die Auswahl zwischen beiden Eingängen.

Der TVP5150 benötigt 2 verschiedene Versorgungsspannungen. Die 1.8V Versorgung wird über einen Spannungsregler aus 5V erzeugt. Die 5V kommen aus einem 12V/5V Step-Down Konverter, der genügend Leistung haben sollte, um neben der Schaltung selbst auch die WS2812 Stripes zu versorgen. Neben der 1.8V Versorgung benötigt der TVP5150 noch 3.3V. Diese werden in obiger Schaltung aus dem STM32F4 Board bezogen. Dabei ist zu beachten, dass auf dem Board eine Schutzdiode (D3) dem auf dem Board befindlichen 3.3V Regler nachgeschaltet ist. Am entsprechenden Pfosten-Pin kommen daher nur ca. 3V an. Da in dieser Anwendung die Schutzdiode überflüssig ist (da das Discovery als Evaluation-Board deklariert ist, kann die Schutzdiode im Experimentierfall ganz wichtig sein), sollte D3 auf dem Board durch eine Drahtbrücke ersetzt werden, oder aber die 3,3V werden über einen zusätzlichen Regler erzeugt.

Zur Bedienung und Parametrierung des Boards gibt es 2 grundsätzliche Möglichkeiten: Der verbaute IR-Empfänger TSOP1036 o.ä. ist zusammen mit der Software in der Lage, IR-Codes mit NEC-Codierung zu empfangen und zu dekodieren. Dieses Protokoll wurde gewählt, um die günstigen IR-Fernbedienungen der üblichen RGB-LED-Stripes verwenden zu können. Solche Bedienungen sind günstig einzeln oder zusammen mit einem RGB-Controller zu kaufen oder sie liegen vielen RGB-LED Komplettpaketen bei.

Das STM32F4discovery Board besitzt 2 USB Schnittstellen. Eine davon ist exklusiv für das eingebaute STLink Programmierinterface reserviert. Die andere USB-Buchse kann jedoch für die Anwendung eingesetzt werden. Die PitschuLight Software implementiert an diesem USB Anschluss einen virtuellen COM-Port (serielle Schnittstelle) und lässt sich somit einfach mit einem Laptop, PC oder auch geeignetem Handy verbinden. Auf dem Laptop/PC muss der entsprechende VC-Treiber (aus dem STLink Package, siehe STM Homepage) installiert werden.

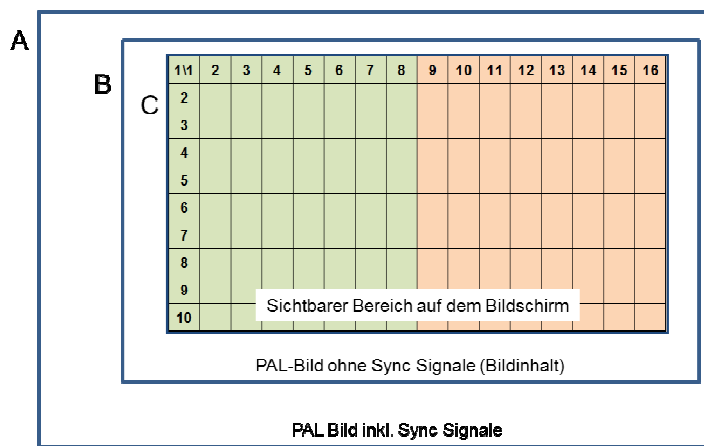
Über diese Verbindung kann dann mit jedem Terminal-Programm (z.B. Putty) mit dem PitschuLight kommuniziert werden. Über die Verbindung gibt das PitschuLight einerseits Informationsmeldungen aus und zeigt Einstellungen an, nimmt über die Verbindung aber auch Kommandos zur Parametrierung entgegen (siehe Kapitel Parametrierung).

Alternativ zur USB Verbindung kann die Kommunikation auch über eine „echte“ serielle Verbindung zwischen PitschuLight und Laptop/PC erfolgen. Hierzu wird die UART3-Schnittstelle (Pins: PD8 = TX, PD9 = RX) mit den Einstellungen 115200 Baud, 8 Bit, 1 Stopbit, No parity verwendet. Die Umschaltung zwischen USB und UART3 Kommunikation erfolgt durch Neukompilierung der Sourcen, wobei die Compilervariable „USE_USB“ definiert sein muss für USB und undefiniert für UART.

Funktionsweise des AmbiLights

In diesem Kapitel sollen einige technische Zusammenhänge und Funktionsweisen des PitschuLights im AmbiLight Modus beschrieben werden. Mit dem besseren Verständnis der Funktionsweise lassen sich leicht Änderungen und Erweiterungen der Software nach eigenen Wünschen anpassen.

Zunächst ein paar grundlegende Rahmenbedingungen die die verarbeiteten CVBS-Signale aber auch der verwendete Videoprozessor TVP5150 mit sich bringen.



Die Abbildung links zeigt den groben Aufbau eines PAL-Signals.

Der **Kasten A** stellt das gesamte PAL-Bild dar. Bilder werden zeilenweise übertragen, wobei jedes sog. Halbbild 312,5 Zeilen hat. 2 Halbbilder gehören zu einem Frame, welcher damit 625 Zeilen hat. Die beiden Halbbilder werden jedoch nicht als 2 Hälften gesendet, sondern ineinander verschränkt, d.h. das erste Halbbild zeigt alle ungeraden Zeilen (1, 3, 5, ...) und das zweite Halbbild die geraden Zeilen (2, 4, 6, ...) an. Damit sind beide

Halbbilder auch einzeln verwendbar, da sie grundsätzlich den gesamten Bildinhalt jedoch mit halber vertikaler Auflösung beinhalten. Im PitschuLight werden aufgrund dieses Umstandes die beiden Halbbilder nicht unterschieden.

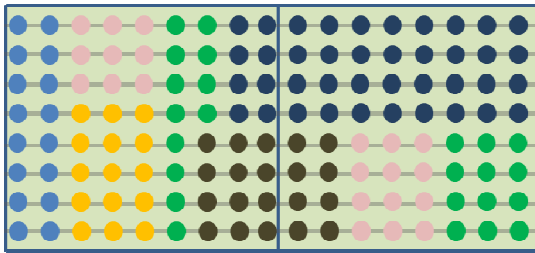
Der Videoprozessor TVP5150 übermittelt den „Kasten A“ in Form eines kontinuierlichen Stroms von Bytes mit einer Übertragungsrate von 27 Mbyte / Sekunde. Pro Zeile werden 1728 Byte mit den Farb- und Helligkeitswerten mit jeweils 2 Byte pro Pixel übertragen. Daraus ergeben sich 864 Pixel pro Zeile. Eine Zeile wird innerhalb von 64 Mikrosekunden übertragen. Bei 625 Zeilen und 25 Bildern pro Sekunde ergibt sich damit die Rate von 27 Mbyte/s ($625 * 25 * 1728 = 27.000.000$).

Der **Kasten B** zeigt den eigentlichen Bildteil des PAL-Signals, also ohne die Informationen, die für die Zeilen und Bildsynchronisierung notwendig sind. Von den 1728 Byte, die eine Kompletzeile darstellen, sind nur 1440 die eigentlichen Bilddaten. Da ja jeweils 2 Byte ein Pixel darstellen, ergibt sich daraus eine maximal sichtbare Bildbreite von 720 Pixeln.

Auch in der vertikalen Richtung sind von den 312 Zeilen eines Bildes einige für die Bildsynchronisation nötig. Der sichtbare Bereich besteht hier aus 288 von den 312 Zeilen.

Der **Kasten C** schließlich zeigt den wirklich sichtbaren Bereich eines konkreten Fernsehers. Dieser Bereich hängt vom Gerät und seiner Einstellung ab. I.d.R. werden die Bildparameter des Fernsehers so justiert, dass möglichst viel vom Bildinhalt sichtbar ist, ohne jedoch Teile der Synchronisationsbereiche in Form von schwarzen Balken zu zeigen. Um das zu vermeiden, wird das sichtbare Fenster bewusst kleiner gewählt als eigentlich nötig wäre. So ist es durchaus möglich, dass ein Fernseher von den theoretisch möglichen 720x288 Pixeln des PAL-Bildes tatsächlich nur 700x270 anzeigt. Für das PitschuLight ist nur der als Kasten C gezeigte Bereich relevant und lässt sich über die Parameter L, W, T und H genau auf die Gegebenheiten des Fernsehers einstellen (siehe Parametrierung).

Aus dem Datenstrom des Videoprozessors wird nun im ersten Schritt ein relativ grob gerastertes Abbild in Form von Blöcken gebildet. Dieses virtuelle Abbild ist immer 64 x 40 Blöcke groß. Im Bild oben ist dies beispielhaft anhand eines 16 x 10 Rasters skizziert. Das folgende Bild zeigt schematisch



2 solcher Blöcke mit den jeweils enthaltenen Pixeln des übertragenen Bildes. Angenommen, der Parameter W ist auf 688 Pixel eingestellt, so werden bei dem Raster von 64 x 40 Blöcken jeweils $688/64 = 10,75$ Pixel einer Zeile zusammengefasst (technisch werden für 4 Blöcke 1 mal 10 und 3 mal 11 Pixel zusammengefasst) und deren Durchschnittswerte für Farbe und Helligkeit abgespeichert.

Weiter angenommen, dass der Parameter H auf 280 Zeilen eingestellt ist, so werden immer 7 Zeilen und von diesen immer die besagten 10 bzw. 11 Pixel zusammen gerechnet. Insgesamt hat ein Block in diesem Fall den Farb- und Helligkeitsdurchschnittswert von $7 * 10$ oder $7 * 11$ Pixeln.

Im Detail läuft dieser Vorgang wie folgt ab:

1. Der TVP5150 wird so eingestellt, dass er das komplette PAL-Bild ohne Randbeschneidung überträgt. Die interne CROP-Funktion wird also nicht benutzt. Die Datenübertragung erfolgt im Modus „8-bit 4:2:2 YCbCr“. Hierbei werden immer 4 Byte-Pakete für jeweils 2 Pixel übertragen. Die 4 Byte enthalten in 2 Byte die Helligkeitswerte (Luminanz) der beiden Pixel und in weiteren 2 Byte die Farbwerte Cb und Cr welche für beide Pixel gelten. Die beiden Pixel haben demnach unterschiedliche Helligkeiten aber immer die gleiche Farbe. Die Output-Pins für die Synchronsignale werden entsprechend den Anforderungen für das DCMI-Interface des STM32F407 gesetzt.
2. Das DCMI-Interface sowie das DMA-Interface werden so programmiert, dass die eingehenden Videodaten per DMA-Transfer in einen RAM-Zeilenpuffer geschrieben werden. Die Register der DCMI-Crop-Einheit werden mit den eingestellten Parameterwerten L, W, T und H befüllt, so dass das DCMI Interface nur den gewünschten Bildausschnitt herausfiltert und in dem Zeilenpuffer ablegt. Das DCMI läuft im „continuous“ mode, d.h. es wird automatisch nach jeder Zeile neu gestartet um die nächste Zeile zu übertragen. Um während der Verarbeitung (Durchschnittsbildung) einer gerade gescannten Zeile bereits die nächste übertragen zu können, wird im DCMI der DoubleBuffer Mode verwendet. Hierdurch werden 2 Zeilenpuffer immer abwechselnd befüllt, sodass man in Ruhe den einen bearbeiten kann, während bereits der andere befüllt wird. Nach diesen Initialisierungen wird das DCMI-Interface gestartet. Mit Beginn der nächsten übertragenen Zeile (HSYNC Signal) beginnt das DCMI-Interface die Daten im RAM abzulegen.
3. Wie bereits eingangs erwähnt, dauert die Übertragung einer Zeile 64us von denen 52us echter Bildinhalt sind. Nach der Übertragung der Bilddaten bleiben demnach nur 12us Zeit um die Bilddaten des Zeilenpuffers nach dem o.g. Verfahren in Blöcke mit Durchschnittswerten zu verarbeiten. Da hierzu viele einzelne Rechenschritte notwendig sind, reichen die 12us nicht aus, um pünktlich für die nächste Zeile gerüstet zu sein. Daher wendet PitschuLight folgendes Verfahren an:
Die Zeilen eines Bildes werden in 2 Hälften verarbeitet, zunächst nur die linke Hälfte, also die grünen Blöcke im obigen Bild (horizontal 1 – 8). Sind alle Zeilen der linken Hälfte verarbeitet, wird das ganze für die rechte Hälfte (rote Blöcke 9 – 16) wiederholt. Um also das gesamte Bild zu analysieren, werden 2 aufeinanderfolgende Bildübertragungen benötigt.
4. Das Ende einer übertragenen Zeile kündigt das DMA-Interface per Interrupt an (DMA2_Stream1_IRQHandler). Nun werden die Daten des Zeilenpuffers gemäß der Rasterung der virtuellen Blöcke aufsummiert. Während dessen werden vom DCMI bereits die Daten der nächsten Zeile in den anderen Zeilenpuffer eingelesen.
5. Sind alle Zeilen eines Bildes in der beschriebenen Weise übertragen und verarbeitet worden, so kündigt das DCMI den Bildwechsel per Interrupt (DCMI_IRQHandler) an. Im entsprechenden Handler wird nun zunächst das DCMI und das DMA Interface für die Übertragung der anderen Bildhälfte (immer abwechselnd linke Hälfte -> rechte Hälfte -> linke Hälfte -> rechte Hälfte ...) vorbereitet und neu gestartet. Ab jetzt bleiben noch mindestens 25 Zeilen à 64us Zeit, die nun vollständigen Blockdaten der eben übertragenen Bildhälfte weiter zu verarbeiten, bevor die ersten Daten des nächsten Bildes übertragen werden müssen. I.W. sind dazu die Daten, welche noch im YCbCr-Farbmodell vorliegen, in das RGB-Modell umzurechnen, um sie später

für die RGB-Leds nutzbar zu machen. Diese Konvertierung von YCbCr nach RGB erfolgt mit Hilfe von einschlägigen Umrechnungsformeln:

$$\text{Rot} = \text{Cr} * 1,403$$

$$\text{Grün} = \text{Cr} * 0,714 + \text{Cb} * 0,344$$

$$\text{Blau} = \text{Cb} * 1,773$$

6. Das Ergebnis der Berechnung ist eine dem virtuellen Abbild entsprechende Blockmatrix (64 x 40) mit durchschnittlichen RGB Werten der jeweiligen Pixel.

Die beschriebenen Schritte laufen völlig unabhängig von der Hauptschleife main() in Interrupt-Handlern ab und stellen damit permanent sicher, dass in der RGB-Blockmatrix immer ein aktuelles, den eingestellten Parametern entsprechendes Abbild vorliegt. Die weitere Aufbereitung der Daten erfolgt in der Hauptschleife in folgenden Schritten. Dabei gehen wir weiterhin von einer Blockmatrix von 16 x 10 Blöcken aus.

1. Sobald der VSYNC Interrupthandler über ein Flag mitgeteilt hat, dass ein neues Bild vollständig vorliegt, erfolgt mit Hilfe der Funktion **ambiLightSlots2Dyn()** die Suche nach schwarzen Balken an den Bildrändern (s.a. Kapitel Parametrierung, Parameter G). Dazu werden zunächst für jede Zeile und jede Spalte der Blockmatrix die Durchschnittswerte für die Helligkeit (= Summe aller RGB-Werte) sowie für den Kontrast (Differenz zwischen dem hellsten und dem dunkelsten Block) berechnet. Für jede der 64 Blockspalten und für jede der 40 Blockzeilen liegen damit Werte für Helligkeit und Kontrast vor.
In den 64 Spaltenwerten wird nun von links und von rechts nach den jeweils ersten Werten gesucht, die über dem durchschnittlichen Schwarzwert liegen oder die einen signifikanten Helligkeitsanstieg anzeigen. Der erste Wert mit diesen Eigenschaften bestimmt somit die äußere Blockreihe (jeweils eine links und eine rechts), die mit den LEDs sichtbar gemacht werden soll. Entsprechend wird mit den 40 Zeilenwerten verfahren, um die obere und die untere Blockreihe, die nicht „schwarz“ sind, zu finden.
Da dieser Schritt bei jedem Eintreffen eines neuen Bildes durchgeführt wird, erfolgt eine permanente Anpassung an evtl. vorhandene schwarze Balken.
2. Nachdem im letzten Schritt die äußeren Blockreihen und Spalten bestimmt wurden, erfolgt in diesem Schritt mit Hilfe der Funktion **ambiLightDyn2Image()** die Aggregation der außen liegenden Blockreihen in eine rechteckige Struktur, deren Höhe und Breite den Parametern X und Y entsprechen (siehe Parametrierung). Die Anzahl der aggregierten Blockreihen gibt der Parameter E an.
3. In diesem Schritt wird durch die Funktion **ambiLightImage2LedRGB()** die rechteckige Struktur mit den Ausmaßen X * Y auf die physikalische Struktur der LEDs mit den Ausmaßen P * R skaliert.
Auf den ersten Blick erscheint die Kette von Schritten (64 * 40 -> X * Y -> P * R) redundant zu sein. Die Zwischenskalierung auf eine Größe von X * Y wurde eingeführt, um die Auflösung des AmbiLights beeinflussen zu können. Die Größe von 64*40 im DCMI nahen Teil ist aufgrund des vorgegeben Video-Timings mehr oder weniger fest. Die physikalische Größe der LED-Kette von P * R ist auch fixiert. Über die Zwischenstufe einer variablen Matrix X * Y von 1 * 1 bis zu 64 * 40 lässt sich die Auflösung des AmbiLights beliebig einstellen.
4. Im dritten Schritt wurden die RGB Werte für die LEDs ermittelt und könnten nun angezeigt werden. Um jedoch Verzögerungen, die im Fernseher bei der Bildaufbereitung entstehen können, auch im AmbiLight zu berücksichtigen, werden die RGB Werte zunächst in einen FIFO-Speicher mit einer Länge, die der Parameter M festlegt, zwischengespeichert.
5. Aus dem FIFO-Speicher werden nun im letzten Schritt die anzuzeigenden RGB-Werte geladen und über eine Timer gesteuerte DMA-Übertragung an den WS2812 LED-Streifen übermittelt und damit zur Anzeige gebracht.