documentation

PitschuLight

A full-screen AmbiLight with up to 312 LEDs on the basis of TVP5150AM1 Video processor and a STM32F4DISCOVERY boards

Status: Version 1.2, November 2014

The photo is from Pat Wat (FrankP)

How it should look if it works

Disclaimer

```
/ ************************************************** ****
 *
 *        Control program for the PitschuLight TV backlight
 *        (C) Peter Schulten, Mülheim, Germany
 *        Peter_ (at) _pitschu.de
 *
 *        Verbatim copying and distribution of this entire source code
 *        in any form is permitted, provided the above notice is preserved.
 *
 *        I put this source code + documentation free of charge and therefore offer no
 *        Support of yet I guarantee for its functioning. also
 *        I assume no liability for the consequences of its use.
 *
 *        The source code may only be used for private purposes and modified.
 *        Any other use requires my approval.
 *
 *        See also: 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 user blue button (mode switch / standby)
 *        07/24/2014        pitschu v1.2 bug fixing in tvp5150_dcmi.c
* /
```

Change history of documentation:

- V1.0 November 2014
- V1.1

Table of Contents

Introduction **4**

Basic structure **4**

Function Description **5**

The Mood Light 6

Constant color 6

Color change of the "seven basic colors" 6

Color change of the DIY (Do It Yourself) Colors 6

Color change in the "Multi-Sinus" mode 7

Operation and parameterization in AmbiLight mode **8th**

Parameterization of PitschuLights by terminal 8th

Description of the parameters 9

Tips on proper setting of the parameters 12

Parameterization via the IR operation 12

The hardware **14**

How the ambilights **15**

introduction

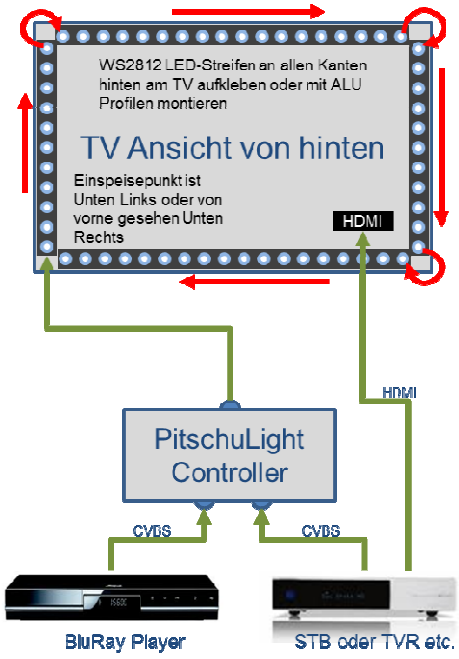
I started in mid-2013, when I swapped m TV. Since the my Philips TV with a new Samsung ingebautes AmbiLight had (but only with HDober Philips one eing left / right distinction), was my sc The very coarse quickly realized that I have a möch AmbiLight again backlightsoute. impact of such a dynamic B the viewing experience is amazing.

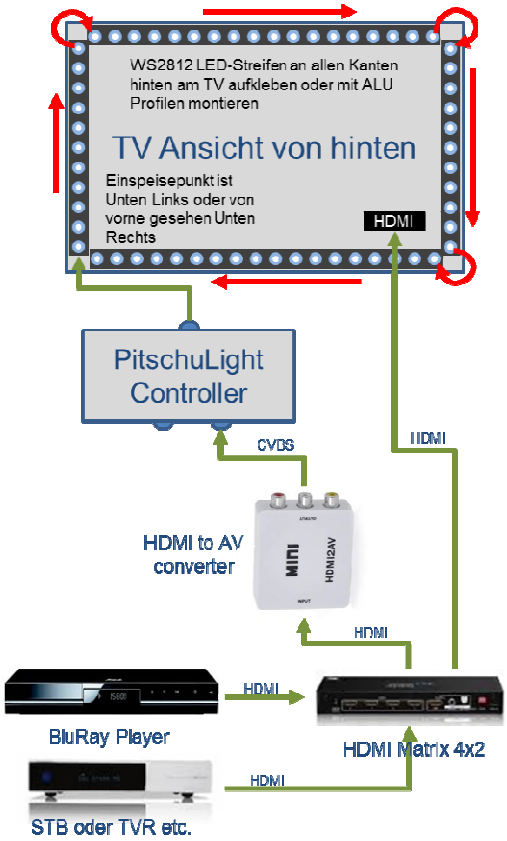
In the market I have no ready Solu on any PC / Linux STB lution found and the replica solutions are based almost st all ftware) addons are not freely available. I wanted to work or appropriate information (schematics and Softw the case a solution thakommt, independently on each and automa ausk with minimal hardware table with the TV / STB off on / off.

Since I had not yet taken a STM32F4DISCOVERY Board, u liegen rd and already some gimmicks so uf basis of contained DCMI had the idea of being able to see. Some research brac interfaces possibly a Lös then pay me to video processor TVP5150AM1 s idea lution Texas Instruments. As a team they are id watching televisi the implementation of a AmbiLight suitable. Since rder run, which also has a 1 I a TV Record need for HDMI processing nich HDMI splitters RCA output, w given layer. Meanwhile, various Varia matrices and CVBSUnterstützung is not necessary yet. t was the variants DMI - tested converters so that a native HDM

basic structure

The following pictures show options used when the n of the basic structure. The left variant kom n devices already a mmt connected G have. The right image shows an opportuni PAL-compatible output (CVBS) ity, rzeugen using a HDMI-matrix and a S) V converter provide such a PAL signal to ores. HDMI2AV while the screen with HDMI





The power supplies for PitschuLight un and LED strips are not shown here.

Regardless of the chosen variant of PitschuLight controller has the task of analyzing video data and convert it into a data stream suitable for a WS2812 LED strip. The LED strip is thereby annularly adhered to the back of the TV, for example, or with corresponding ALU profiles, which distribute the light beautifully mounted. The beginning of the strip is seen from behind Down / Left, then goes up, then bends and so on until it reaches the bottom left to the right. In the corners, it is simply severed and moves the cores 3 of the strip with short cables around the corner to the next track. The length of LED strip 4 is adjustable between LED 312 and LEDs (312 = 2x60 2x96 side edges plus top and bottom) in the PL. The lower strip can also be omitted, for example if the cable stuff etc.

To determine the correct number of LEDs, one should measure the glass window of the TV and cut correspondingly long pieces of WS2812 Strip. Possibly. must be a little shorter, because the mounting options on the back require the. Minor discrepancies between dimensions of the screen and the length of the strip is not tragic, with larger discrepancies but you can already see any significant shifts between image content and LEDs.

function Description

The PitschuLight can be operated in two different modes:

1. in AmbiLight mode, the LEDs are driven based on the image information of an input video signal. For this purpose, the video image is analyzed and projected both color and brightness of the image outer edge to the corresponding LEDs. So the TV picture will be expanded into the emptiness in all optical.

Second The Mood Light mode, the LEDs are driven independently of an input signal that is set either a constant color or temporally changing colors. In further variants are also different, time varying patterns can be displayed.

The PitschuLight is switched via the red ON / OFF button on the remote control or alternatively via the blue button on the STM32F4DISCOVERY board an ON and OFF. To one of the aforementioned button must be pressed briefly. When switched off, the LEDs are simply no longer controlled and remain dark. Still active, however, the automatic channel search of ambilights remains (if the parameter V = video source activated). In this case, the LED in the left and right lower corner indicate is searched on which channel just after a video signal. Is on any of the channels, a video signal is present, switch the two LEDs in 2Sekundenrzthmus. When switched on, the LEDs are driven in accordance with according to the active mode (or AmbiLight Mood Light).

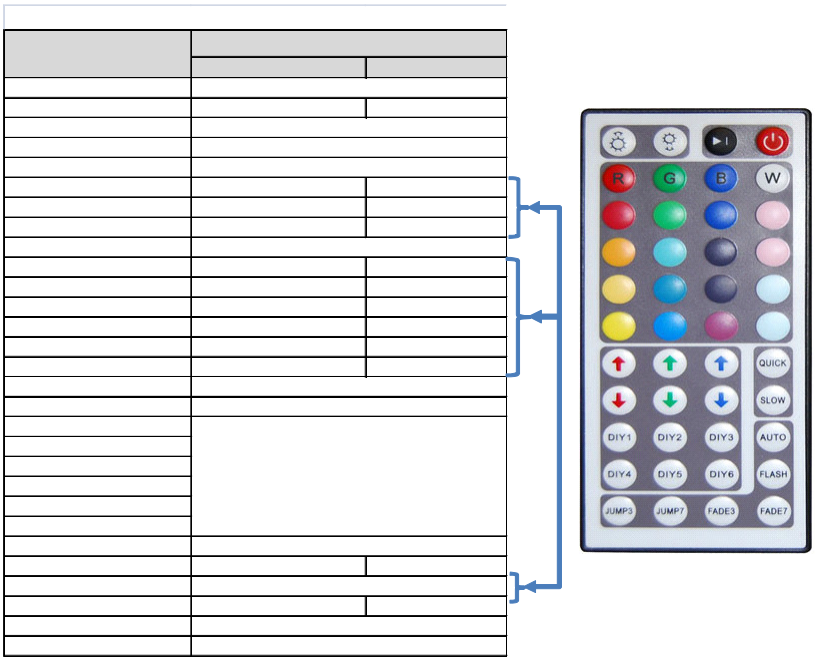
Switching between AmbiLight and Mood Light mode is done by pressing the red ON / OFF button for more than 2 seconds. The mode remains active until you return to the ON / OFF button is pressed for more than 2 seconds.

The available on the Board STM32F4DISCOVERY 4 LEDs (red, blue, green and orange) are used as a function control for various AmbiLight operations. They indicate the following states:

- The green LED indicates receive activity on the IR receiver, ie it should flicker when a key is pressed on the IR operation.
- The red LED indicates whether the horizontal sync signal (HSYNC) from the video processor is active.
- The orange LDE indicates whether the vertical synchronizing signal (VSYNC) from the video processor is active.
- The blue LED is switched at different time points within the main loop of the program on and off. You can use it for time measurements with an oscilloscope.

The Mood Light

As an alternative to the PitschuLight AmbiLight mode can be used as Mood Light (= mood light). The following figure shows the assignment of FB buttons in the Mood Light mode and is used to better understand the then following description of the Mood Lights:



To activate the Mood Lights OFF button for more than 2 seconds is operated to switch from AmbiLight in the Mood Light mode, the ON /.

The basic brightness of the mood lights is adjusted by the two brightness buttons (top left of the FB). These keys are independent of the selected mode.

The Mood Light can basically be operated in the following modes:

constant color

To this end, a color button is pressed, after which this color is displayed after the set fade time. The fading time can be on the keys QUICK / SLOW shortened (= rapid color change) or extended (= slow color change, up to 30 seconds). The respective RGB can be changed value of the displayed color and thus any color are set using the red, green and blue arrows. A long press (> 2 seconds) one of the 6 DIY (Do It Yourself) keys will be stored on one of the six available memory locations the color currently displayed. Briefly press DIY key then represents the stored color.

By very long press (> 4 seconds) a DIY button will delete the stored including color back and emptied the DIY store.

Color change of the "seven basic colors"

With the key FADE7 which are run through "7 colors" in the fixable by the key QUICK / SLOW fade time.

Color change of the DIY (Do It Yourself) Colors

With the key FADE3 the stored among the DIY key colors are run through in the definable by the key QUICK / SLOW fade time. The fading time can be on the keys QUICK / SLOW shortened (= rapid color change) or extended (= slow color change, up to 30 seconds).

In the section "constant color" (see above) is described the determination of the DIY colors. If not all DIY store are assigned a color, the color change takes place via correspondingly less DIY colors.

Color change in the "multi-sine wave" Mode

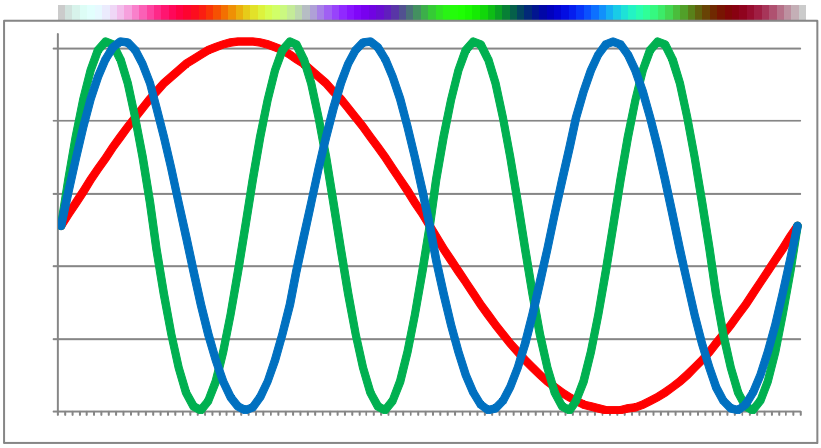
With the key Jump3 the so-called. Multi-sine mode is started. This mode provides a virtually unlimited variety to present different patterns, moving effect lights or running effects. Basically, thereby the 3 primary colors of the LEDs are controlled by configurable sine swing, wherein the amplitude and background brightness can be set in addition to the period time. Below the relationships are to be briefly described.

Assuming that the LED strips consists of 100 WS2812 LEDs. Each of these LED has a red, a green and a blue sub-LED. In the multi-sine mode, the red portion 100 LEDs are driven with a brightness curve sínusförmigen now, in which both the frequency and the amplitude of the background brightness are adjustable. Separated "red" of the curve there are corresponding curves for the 100 green and 100 blue part LEDs. The following table shows these nine parameters in the overview to their default values.

	Key Red	Green Blue		
frequency	G	1	4	3
amplitude	B	127	127	127
brightness	W	128	128	128

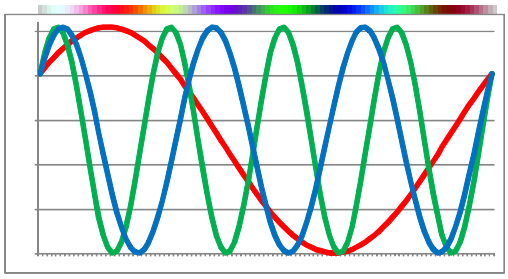
The change of values over the corresponding color-coded arrow keys. In order to change, for example, the amplitude values, the blue color key is first pressed with the printed B. Then the values of each color can be adjusted via the 6 keys up or down. A green LED on the top of the LED strip shows the current value as a percentage of (left = low, right = high).

The following figure shows schematically the corresponding table values controlling the LEDs:



At the top of the image is then visible, together mixed color of the LEDs is shown. the respective red / green / blue values of the sub-LEDs are plotted for each of the 100 LEDs. The technically possible values of each LED color range from 0 (= dark) to 255 (= full brightness). So the initial brightness of 128 is at half

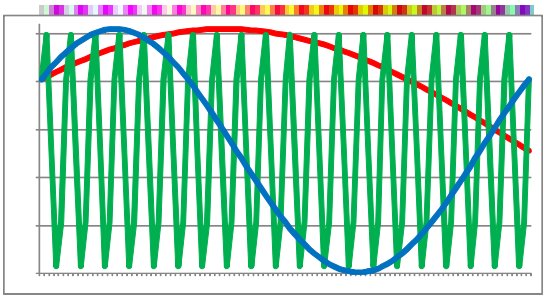
Maximum modulation (= center of the graph). The frequency of oscillation ROT is 1 over all LEDs 100, GREEN and BLUE vibrations corresponding to 4 3 cycles. The number of oscillations can be adjusted from about 0.2 to about 44th



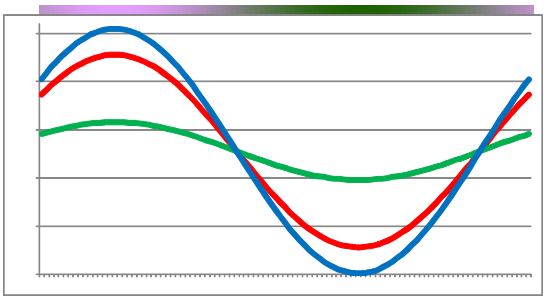
is further set the lead time for a constantly changing phase QUICK / SLOW keys. The phase one goes through the entire image within the set cycle time. It can be from 0.25 seconds to 30 seconds. The picture shows the view as above, but with an altered phase. The curves no longer start at the first LED, but are shifted. By constantly

changing phase so formed a fine movement effect which the light pattern along the LEDs

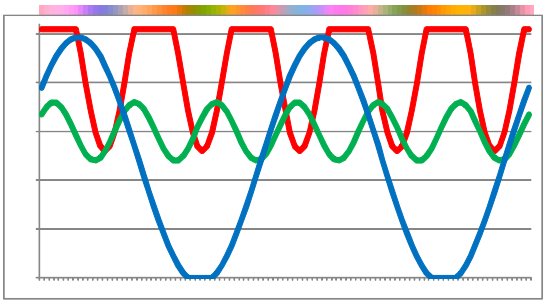
shifts.
By any change in the parameters mentioned completely different illumination patterns can be produced. The following pictures show some examples:



	Key Red	Green	Blue	
frequency G		0.4	20	1
amplitude B		127	127	127
brightness	W	128	128	128



	Key Red	Green	Blue	
frequency G		1	1	1
amplitude B		100	30	127
brightness	W	128	128	128



	Key Red	Green	Blue	
frequency G		5	6	2
amplitude B		100	30	127
brightness	W	230	150	120

Once a set combination of values may be assigned and stored by a long press (> 2 seconds) of one of DIY keys. the combination of parameters is loaded and used to display by pressing the DIY key.

Operation and parameterization in AmbiLight mode

The operation of the PitschuLights in AmbiLight mode using either the terminal connection or via the IR remote control.

When operating as MoodLight operate exclusively via the IR operation.

Parameterization of PitschuLights per terminal

The parameterization of PitschuLights is best done via a terminal connection, since the exact settings of the changed variables at the terminal are visible in this case. A parameter setting via IR operation is possible, but a feedback about the new settings take place only via a green LED on the top edge of the TV.

The following list shows the supported commands in terminal mode:

```
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 = width virtual image 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 = showInfo about Dyn matrix N = restart TVP5150 and show info reg 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

Basically, a value is to parametrierender festgelgt initially with a letter (large or small). This value is then by repeatedly pressing the '+' or '- increased by one button or decreased by one. If, instead, + 'or' -, the letter 'd' is actuated, the parameter is set to its default value. After each one of these buttons to confirm the new value of the parameter is displayed.

Approximately 8 seconds after the last change in a parameter of the current state of the parameter is written in the FLASH memory of the STM32F407 so that the parameters are, even after power is turned off when switching on again available and used. Therefore, after changing a parameter, the supply should be interrupted immediately, so that the storage process can still be carried out.

Description of Parameters

Below all setting parameters and their impact in PitschuLight mode are briefly described. Following the summary, the MIN, MAX and default values of the parameters are given.

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

These 4 parameters reflect the homonymous register values of the TVP5150. The adjusted values are transferred directly to the register. Details of these parameters can be drawn from the data, otherwise it should be clear, since it also

eg in the setup of the TV usually gives these four parameters.

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

P = Physical imagewidth: # of LEDs R = Physical imageHeight: # of LEDs

These parameters determine how many LEDs are physically connected. They must correspond exactly with the number of LEDs on the horizontal and vertical LED strip.

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

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

These 4 parameters determine which image section of the television should be displayed with the LEDs of the PitschuLights. Ideally, these values are set so that
for example a television picture showing in all 4 corners of a limited, colored area, so appears exactly on the LED strip ie the 4 corners of the LED ring show exactly the same colors as the TV. Are the four parameters are set incorrectly, resulting

eg shifts of LEDs or the LEDs do not indicate the color areas again.

Technically, these parameters define the so-called. Crop window (see operation), so the section of the given PAL Windows, which should be displayed. An image line of the PAL signal has 864 pixels, but of which only 720 visible content.

A concrete TV cuts thereof usually something else, so that really visible range, for example, starts at 80 pixels and is only 700 pixels wide. These two values are set accordingly. For vertical adjustment applies analogously. For tips on setting, see the relevant section below.

Para MIN MAX DEF		
L	40 200	80
W	200740696 T	
	4150	16
H	40312288	

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

This parameter determines the speed of adjustment of PitschuLights can be adjusted to the image content. For large values, the adjustment for small values is very fast, very slowly. So the PitschuLight reacts at a setting of 128 directly to image changes, which can be very disturbing for rapid image sequences. For small values, a leisurely adjusting the PitschuLights to the image content, which may be perceived as too slow results. The default setting of 32 is a good compromise.

Para MIN MAX DEF		
I	1 128	32

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

The PitschuLight understood as an extension of the picture to the outside and receives, the outer image content as the default. This can only be the outermost rim (setting = 1) or a wider outer edge are added whose image contents as mixed values to the PitschuLight example. Technically, this parameter defines the number of virtual blocks that are mixed and passed on to the PitschuLight (see the functional description).

Para MIN MAX DEF			
e	1	11	4

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

The PitschuLight adapts permanently to the visible image area on the television. This image area can be very different by different format adjustments and -skalierungen in various TV content. The PALFormat used by PitschuLight has an aspect ratio of 4: 3. Today usual TV usually use the 16: 9 format and image content such as movies using the format eg 5: 3rd The formats are scaled either in height or in the Breits to represent the content or it is only an image section really visible. Depending on the scale it may on the screen to black areas on the left and right or up and coming down. Since the PitschuLight uses the outer image area for controlling the LEDs, including the LEDs would remain always dark at the top, if a movie with 5: 3 format on the screen with black bars shown above and below. To avoid searching the PitschuLight permanently to black areas on all four sides, thus dynamically adjusts the four pages to the "illuminated" image area.

Via the parameter "G" to influence the speed of adjustment of the search functions. In G = 0, the mechanism is turned off so that no adjustment takes place. This value is actually only used, while other parameters (L, W, T and H) are set. Larger values 0 specify the number of frames (Transferred images), whose contents are averaged and used to determine the beam. A value of 200 means, therefore, that over 200 frames received (about 8 seconds) in the calculation. With large values, for example, prevents a dark scene, the image borders are anyway almost black, does not result that the PitschuLight immediately adjusts it and might play a more lying in the center part of the image in the LEDs. This would disturb the overall impression strong.

Through the software, the search area by black bars to approximately ¼ of the image height and about 1/6 of the image width is limited.

Para MIN MAX DEF	
G	0200100

X = width virtual image in blocks Y = virtual image height in blocks

The PitschuLight works internally with a rough image of the screen. This virtual image is used around the individual calculation steps (sa operation) from the video signal to the LED driver to perform. Ultimately, these two parameters define the resolution of the virtual image and hence the resolution of the LEDs. The maximum values of the parameters are X = 64 and Y = 40. This means that, for example, may represent up to 64 different areas of the LEDs on the upper and lower edges, even if more LEDs are installed. If, for example, on the other hand, the values X = 2 and Y = 2, a so distinguishes the PitschuLight only respective regions 2 in X and Y direction. One sees, for example, that light up at the top all the same, the left half of the LEDs and reflect the color value of the left half.

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

After selecting the parameter "V", via the keys "1" and "2", the corresponding input terminal are selected as the source. By selecting "0", the PitschuLight is placed in the scan mode. Here, the PitschuLight permanently searches on both inputs for a signal. Once a signal is found, the signal to the LED drive is used. If the signal (source, for example, is turned off), the PitschuLight returns to the scan mode.

Between the inputs are not distinguished, ie there is no operating prioritization of one of the channels (first come first serve).

Q = showInfo about Dyn matrix

This is not a parameter, but merely a debug function of PitschuLights. It issued the contents of the search variables of the dynamic "black bar adjustment" on the terminal.

N = restart TVP5150 and show reg info

As the name suggests, the video processor TVP5150 is initialized to the reset values are reset and new. Then all the stops TVP5150 be issued at the terminal.

A = set TVP5150 auto gain control ON / OFF

The video processor TVP5150 has a built-in function for automatic adjustment to the image signal. This ensures, for example, that a weak image signal is sufficiently amplified to operate the clean PitschuLight. On the other hand, it can also cause dark areas appear too bright in the LEDs.

said fitting function ON and OFF This parameter can also be switched.

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

Modern televisions prepare for the incoming video signals through more or less complex image processors and display them. The time for processing the signal for LCD / plasma screen takes some time, ie when the signal includes a sudden brightness or color change that is visible only after several milliseconds on the screen. The PitschuLight is able to represent this color and brightness changes almost in real time with no delays at the LEDs. To such a change to synchronize, so to let the LEDs and the screen appear the same time, a delay time can be set in PitschuLight this parameter. This delay time can be between 0 (= no delay) and up to 20 frames (about 0.8 seconds). The parameter should be set to

Para MIN MAX DEF

N	**0**	**20**	**0**

Tips on proper setting of the parameters

Below all the relevant setting parameters and their impact in AmbiLight mode are briefly described (for description of the parameters above).

1. First set the number of existing LEDs with P and R.
2. The X and Y parameters should have the same values as P and R. Larger values do not add value; smaller values make the LED display coarser (multiple LEDs with the same color).
3. To adjust the left / right or top / bottom limits (L / W / D / H) the dynamic beam detection should be turned off, so set the parameter G to 0.
4. The parameter E should be the first Thus, the LEDs indicate only the respective outermost strip of the image.
5. The margin setting (parameter L, W, T and H), preferably with a still image (ideally Comics) make. set the limits so that the LEDs indicate the colors of the visible outer edge of the picture. The image must be full-screen, so can have no bars. The following procedure is recommended:

    O L set very small and then locate the left edge by L is increased.
    Here you have to pay attention to the left-hand LEDs on the top and bottom strips. If LEDs are missing or weak flicker, then the value is too small. Afterwards W very large set the parameters and seek the right edge by decreasing W. Here, too, you have to pay attention to the right LEDs above and below. Missing or LEDs flicker weakly, then W is still too large.

    O For the up / down adjustment T analogous to set very small and then the
    Search the top by T is increased. Here one pays attention to the upper right and left LEDs. Then set H very large and seek the bottom by H is slowly reduced again. Here one pays attention to the lower LEDs left and right.

6. After setting the limits, the parameters G and E should be reset (see parameter description).

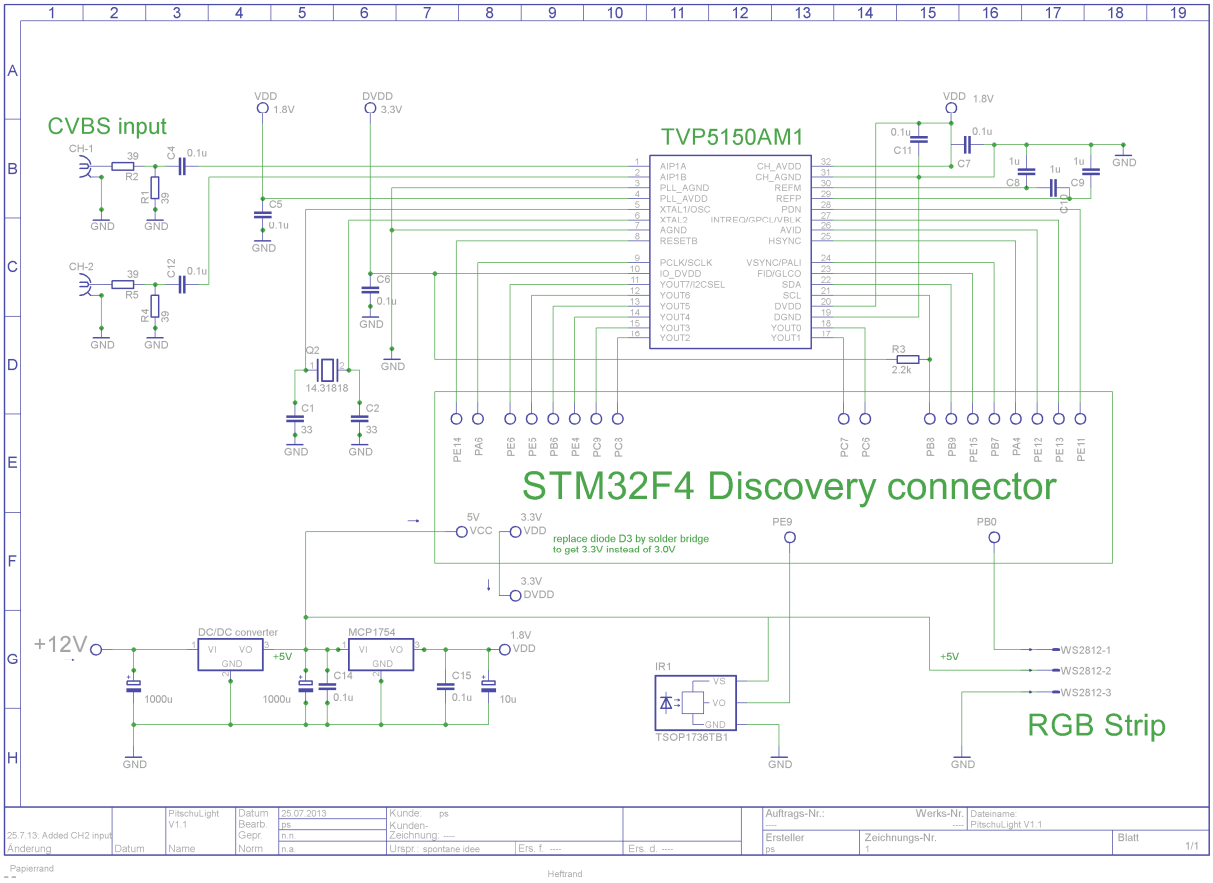
Parameterization via the IR operation

Via the IR remote control most of the PitschuLight can be set parameters for the AmbiLight mode (description of the parameters see previous chapter). In the Mood Light mode, the operation is performed via the IR remote control. To describe the various light treatments see "Mood Light".

As can be performed any issue of current parameter values in the IR control (as in terminal mode), the LED is used as a display strip of the top instead. Here, an LED lights up in green color with maximum brightness. The leftmost LED indicates a value of 0%, the rightmost value of 100%. The percentages are based on the respective setting of the currently edited parameter value.

The following table shows the assignments of all the keys of the 44-Key used IR remote control in the two operating modes of the PitschuLight.

The hardware



The picture shows only the circuit board in addition to STM32F4DISCOVERY needed. The STM board has 2 pin connectors on an additional board with the components shown is plugged. The wiring of the TVP5150 corresponds to the standard connection specified in the data sheet. 2 Chinchbuchsen CVBS signals (also called CVBS) eg from the TV, TV recorders and Blu-ray players will be recorded. An automatic or manual switching of channel TVP5150 enables the selection of two inputs.

The TVP5150 requires two different supply voltages. The 1.8V supply is generated by a voltage regulator of 5V. The 5V come from a 12V / 5V step-down converter, which should have enough power to supply the WS2812 Stripes next to the circuit itself. In addition to the 1.8V supply to the TVP5150 still 3.3V required. These are based in the above arrangement of the STM32F4 Board. It should be noted that on the board, a protective diode (D3) is connected downstream of the on-board 3.3V regulator. Therefore, at the appropriate post-Pin only about 3V arrive. in this application because the protective diode is redundant (since the discovery is declared as evaluation board, the protective diode in the experimental case can be quite important), should be replaced D3 on the board by a jumper, or the 3,

For operation and parameterization of the board there are two basic ways: The built IR receiver TSOP1036 etc. together with the software is able to receive IR codes with NEC encoding and decoding. This protocol was chosen to use the favorable IR remote controls the usual RGB LED Stripes. Such operations are individually favorable or purchase along with a RGB controller or they are many RGB LED complete packages at.

The STM32F4DISCOVERY board has two USB interfaces. One of which is reserved exclusively for the built STLINK programming interface. However, the other USB port can be used for the application. The PitschuLight software implemented on that USB port a virtual COM port (serial interface) and can therefore be connected easily with a laptop, PC or suitable mobile phone. On the laptop / PC, the corresponding VC driver must (from the STLINK package, see STM website) to be installed.

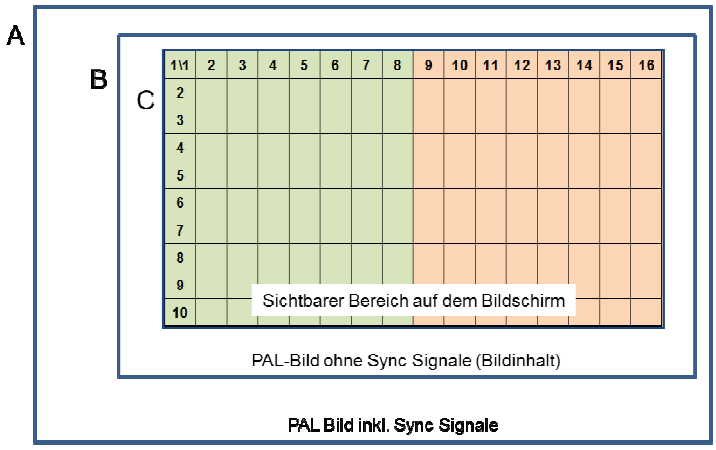
can then communicate with each over this connection. Other the terminal program (eg Putty) to the PitschuLi clothes are the PitschuLight light the connection and displays settings takes practice one hand Informationsmeldun about the connection but also commands fo ungen parameterization contrary (see section Pa configuration).

Alternatively, the USB connection can co between rfolgen communication via a "real" serial conn. For this, the UART3 interface ce ention 8 = PitschuLight and laptop / PC erfo PD9 = RX) with the (Pins is: PD8 0 baud, 8 bits, 1 stop bit, no parity used The three TX, settings 115200 B switching between USB and UART3 k communication is done by recompiling the Sou must be defined for USB and where the compiler variable "USE_USB" de undefined for UART.. ourcen,

How the Amb biLights

In this chapter, some technical in AmbiLight mode will be e relationships and modes of operation of the PitschuL let uLights described. easily changes and enhancements Adjust With the better understanding of how the software en la according to your wishes n..

First, a few basic Rahmenb used video processor with nbedingen the processed CVBS signals but it is au bring. also the TVP5150



The figure on the left shows the great robes development of a PAL signal.

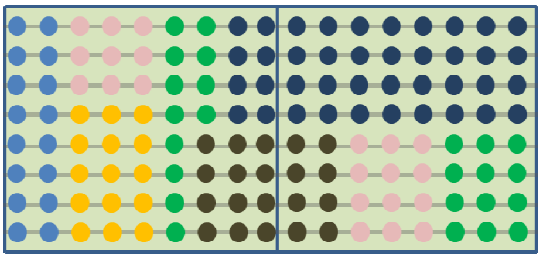
Of the **Case A** represents the whole image. te PALeise Therefore, images are transferred zeilenwei lbbild each so-called. Ha include it has 312.5 lines. 2 fields ge to a frame, which 625 th he therefore has 6 rows. The two fields are not transmitted as 2 halves, but intertwined, ie, the first Halbb shows all the odd lines (1, 3, and the second field, the screened lines (2, 4, 6, ...) in. Are thus lbbild . 3, 5, ...) raden ind both half en Fields can be used individually, as they include a vertical but they are basically huLight the entire picture content with ha, given such resolution. In Pitschu fields did not differ. circumstances, the two

The video processor TVP5150 transmitted d bytes at a LT the "box A" in the form of a continuous stream Mbyte / second. 1728 bytes es of s transfer rate of 27 M and brightness values, each with 2 are transferred with the per pixel per line. This results in transfer 864 pixels Farbpro ren bytes per line. A line is within 64 Mik second, this results per ikrosekunden. With 625 lines and 25 frames 27 Mbyte / s (625 * 25 * 1728 the rate of 27 = 27,000,000).

Of the **Case B** shows the actual Bildte lines and picture dteil of the PAL signal, ie without the information f dig. Of the 1728 bytes, ie for synchronization represent necessary, the actual B are co mplete line of image data n. Since 2 bytes each representing a pixel, erg 1440 this results in a maximum visible Bildbre also in the width of 720 pixels. results vertical direction are needed by de. The visible range is here au to 312 lines of a picture, some of the vertical sync from 288 of the 312 lines es isation

Of the **Case C** finally shows the real range depends on the the visible region of a specific TV. The position from. Usually, the image ieser seer device and its Einste adjusted so that as much as possible parameters of the Television set nhalt is visible, but without showing part of ns Fenster areas of Bildinha in the form of black bars the theoretically the synchronization. To avoid this, the visible Fe would tig. So it is quite rather possible 720x288 pixels is only as a box parameters L, W possible that a TV In the PAL image actually shows only 700x270. en C range For the PitschuLight to deliberately chosen smaller than shown relevant and can be adjusted by the conditions of the TV (see necessary, T and H exactly the Ge parameterization).

formed from the data stream of the video processor images now will be first step, a relatively coarse rasterized ses virtual image is tes image in the form of blocks. this is this above the example of a always 64 x 40 blocks in size. In B 16 outlines x 10 grid. The following figure matically shows schem ltenen



2 such blocks with each containing pixels of the transformed image. Angenom the parameter W is 688 pixels so einge stellt, be in the grid of 64 x 40 B respectively 688/64 = 10.75 blocks pixels of one row together (technically be summarized for blocks 1 and 3 times 10 times 11 pixels) and the Average Values abgespeich for color and brightness ÜR 4

hnitts secured

Next, assuming that the parameter and of these always is is set to 280 lines H, as always 7 Z zw. 11 pixels are added together. Overall Lines 10 or in this case, the color and Helligkeitsdur a B has urchschnittswert of 7 * 7 * 10 or 11 pixels. n block

In detail, this process is as follows: b:

1. The TVP5150 is set to d transmits. The internal CROP funct mode "8-bit 4: 2: 2 YCbCr". Hier transmitted. The 4 bytes contained in the 2 bytes and in further Farbwer thus have different H Outpu the pins for the Synchronsig DCMI interface of STM32F407 g

   It that he will nction the full PAL image without Randbeschneidu so do not use. The data transfer at erf always be 4 byte packets for every 2 pixels in 2 bytes, the brightness values (luminance) of the two P erte Cb and Cr which are common to both pixels. Both e brightness but always the same color. signals are set according to the requirements for there. is ung done in Pixel identical Pixe the

2. The DCMI interface and the DM video data via DMA transfer in the e DCMI crop unit with so that the DCMI interface deposits only de line buffer. The DCMI Ramparts line restarted at the next (averaging) a straight can be filled alternately in DCMI doubleb so that the other is being filled. After these initializations transferred line (HSYNC Signa will store.

   interface are programmed are written Incoming a RAM line buffer. The Reg with the set parameter values L, W, T, and H be the desired image, and filters out in the äuft in "Continuous" mode, ie it will automatically transferred to ste line. To already transferred the next during processing ade scanned line to line Buffer mode used. In this way you can be ass edit the one in peace 2 Zeilenpu while b

   before ends register filled ch each to to be buffered d already

   around the DCMI interface starts. With the start of the next nal) the DCMI interface starts the data in RAM th

3. As mentioned at the outset, dau real image content are. process the image data of the line buffer data after Ov. Since the object of many individually order in time for the next Z methods:

   modified transmission of a line of which 64us 52us transmission of the image will therefore only 12us Z according to the abovementioned process in blocks with Durchschnittswer elne calculation steps are necessary, which 12us ne line range to be prepared. Therefore applies PitschuLight success

   The lines of an image are in 2 green blocks (in the image hori is to analyze the whole for the right Hälft be 2 aufeinan

   processed into 2 halves, initially only the left half, as orizontal 1-8). If all lines of the left half verarb half (red blocks 9 - 16) are repeated. So, to require the other GESAM following image transfers. hurry announces the DMA interface via Interupt to now, the data of the line buffer, according to the summed. Meanwhile be read by DCMI prepared other line buffer. he transferred worde described manner and processed to sel via Interrupt (DCMI_IRQHandler). In the match DCMI and the DMA interface for the transfer of at inke half -> RH -> left half -> right Hälf b remain still at least 25 lines of 64us time, d s transmitted image half to further process before s must be transferred , IW are to present the data dell, converted in the RGB model to them

   So the rbeitet, lonely image

   time to Erten to not olgendes Light success

   already the

4. The end of a transmitted line (DMA2_Stream1_IRQHandler). Nu screening of the virtual blocks au of data of the next line in the

5. Are all lines of an image in announcing the DCMI the Bildwechse handler the DC half of the screen (always will be alternately link prepared and restarted first. From each complete block of data just the first data of the next image which still in the YCbCr color model

   lems, to end other älfte ...)

   That now or die th, he later

to make use of RGB LEDs. This conversion from YCbCr to RGB is performed using relevant conversion formulas:

Red = Cr * 1,403
Green = Cr * 0.714 + Cb * 0,344 Blue = Cb *
1.773

6. The result of the calculation is a virtual image corresponding to the block matrix (64 x 40) with average RGB values of the respective pixels.

The described steps are completely independent of the main loop main () in Interupt handlers, and so provide permanent sure that in the RGB block matrix is always present a current, corresponding to the set parameter image. The further processing of data occurs in the main loop in the following steps. We continue to expect a block matrix of 16 x 10 blocks.

1. Once the VSYNC interrupt handler has informed about a flag that a new image is completely present, is done with the function **ambiLightSlots2Dyn ()** the search for black bars at the edges (see Chapter parameterization, parameter G). For this purpose the block matrix is first calculated the average values of the brightness (= sum of all RGB values) as well as for the contrast (difference between the lightest and darkest block) for each row and each column. thus we have values for brightness and contrast for each of the 64 block columns and for each of the 40 block rows.

   In the 64 column values will be searched from the left and from the right to the respective first values that are above the average black level or display corresponding to a significant increase in brightness. The first value with these properties thus determined the outer block row (one left and one right) to be visualized with the LEDs. A similar procedure is the 40 row values to find the upper and lower row of blocks that are not "black".

   Since this step is performed at each of a new image arrives, then a permanent adaptation to any existing black bars.

2. After the outer block rows and columns have been determined in the last step is carried out in this step with the aid of the function **ambiLightDyn2Image ()** the aggregation of the rows of blocks located externally in a rectangular structure, whose height and width X and Y correspond to the parameters (see configuration). The number of aggregated rows of blocks indicates the parameter e.

Third In this step, by the function **ambiLightImage2LedRGB ()** the rectangular structure with the dimensions X * Y scaled to the physical structure of the LEDs with the dimensions of P * R.

the chain of steps appears at first sight (64 * 40 -> X * Y -> P * R) to be redundant. The intermediate scaling to a size of X * Y was introduced in order to influence the resolution of the ambilights. The size of 64 * 40 in DCMI near part is due to the specified video timings more or less fixed. The physical size of the LED chain from P * R is also fixed. Via the intermediate stage to a variable matrix X * Y of 1

* 1 up to 64 * 40, the resolution of ambilights can be set arbitrarily.

4th In the third step, the RGB values were determined for the LEDs and could now be displayed. However, delays that may arise in image processing in the television, to be considered in AmbiLight, the RGB values are first temporarily stored in a FIFO memory with a length that determines the parameters M.

to be displayed RGB values 5. From the FIFO memory now in the last step is loaded and controlled by a timer DMA transfer to the WS2812 LED strips transmitted and displayed.