Application Note

Description of RGB Remote SDK Functions

AN-19120602-E1

Version 1.0.0

2019-12-06

Key Words:

RGB Remote, SDK Function, Key Matrix

Brief:

This document provides the description for RGB remote SDK functions.

TELINK SEMICONDUCTOR

Published by

Telink Semiconductor

Bldg 3, 1500 Zuchongzhi Rd,

Zhangjiang Hi-Tech Park, Shanghai, China

© Telink Semiconductor

All Right Reserved

Legal Disclaimer

This document is provided as-is. Telink Semiconductor reserves the right to make improvements without further notice to this document or any products herein. This document may contain technical inaccuracies or typographical errors. Telink Semiconductor disclaims any and all liability for any errors, inaccuracies or incompleteness contained herein.

Copyright © 2019 Telink Semiconductor (Shanghai) Ltd, Co.

Information:

For further information on the technology, product and business term, please contact Telink Semiconductor Company (www.telink-semi.com).

For sales or technical support, please send email to the address of:

telinkcnsales@telink-semi.com

telinkcnsupport@telink-semi.com

Revision History

Version 1.0.0 (2019-12-06)

This is the initial release.

Contents

1. Overview

RGB remote consists of a 5*3 key matrix and a LED. It is used to scan keys and check if it's long press according to key values.

After the remote is powered on, RF initialization and IO initialization start first. The system reads the analog register 0x3b as serial numbers of packets, and writes the serial numbers of packets to 0x3b before deepsleep. The analog register value keeps unchanged after deepsleep wakeup. After initialization, the remote scans key values and returns list values by key values as indexes. If the return values are commands of chroma or luminance adjustment of color temperature (CT) LEDs, then the serial numbers of packets sent every 320ms will increase by 1, which means trigger the command again. If no key is pressed within 200ms, the remote will enter deepsleep.

2. Description of Data Structure

2.1 RF Packet Format

```
typedef struct{
    unsigned int dma_len;
    unsigned char rf_len;
    unsigned char rf_len1;
    unsigned short vid;
    unsigned int pid;
    unsigned char pkt_seq;
    unsigned char key_control;
    unsigned short value[3];
}LED_Package_t;
```

dma_len: RF is in DMA mode, dma_len represents the length of a packet, excluding dma_len.

rf_len: If the communication mode is private 2.4G, the length of data, rf_len = dma_len-1; if it's BLE mode, rf_len as header information can be defined by users.

rf_len1: If the communication mode is BLE mode, the length of data, rf_len1 = dma_len-2; if it's private 2.4G mode, rf_len1 as user data can be defined by users.

vid: ID of product types. IDs can be defined by users according to different products.

pid: Product ID. Every remote has its unique ID.

pkt_seq: Serial number of data packets. Once a command is sent by a remote, the serial number will increase by 1 automatically.

key_control: Control command value. The command values are as follows:

```
typedef enum{
    KEY_NONE_CMD=0, // Null command
    KEY_ON_CMD, // Key on command
    KEY_OFF_CMD, // Key off command
    KEY_LUMINANCE_INC_CMD, // Luminance increase command
    KEY_LUMINANCE_DEC_CMD, // Luminance decrease command
    KEY_CHROME_INC_CMD, // Chroma increase command
    KEY_CHROME_DEC_CMD, // Chroma decrease command
    KEY_SET_CHRO_LUMI_CMD, // Set chroma and luminance command
    KEY_NIGHT_CMD, // Nightlight command
    KEY_PAIRE_CODE_CMD, // Pairing command
    KEY_CLEAR_CODE_CMD, // Memory clearance command
    KEY_SET_RGB_CMD, // Set RGB LED command
    KEY_BREATH_RGB_MODE_CMD, // RGB breath mode command
    LED_LAST_CMD,
}LED_Control_CMD_e;
```

Value: When it's a command of setting luminance of RGB LEDs, the value indicates the luminance of RGB LEDs, the maximum value is 1000. Value[0], Value[1] and Value[2] indicate the luminance of the red, green and blue LEDs respectively. When it's a command of setting CT LEDs, Value[0] indicates the index of luminance, Value[1] indicates the index of chroma.

3. Description of Functions

void gpio_init_func(void)

Function: GPIO initialization

Parameter:

Return Value:

Note: Set the 5*3 matrix keys to input state, set input ports as internal 1M pull-up resistor, set output ports as floating in idle state, set high/low level by internal pull-up/down resistor when scanning keys.

void rf_init_func(void)

Function: RF initialization

Parameter:

Return Value:

Note: Set RF address, buffer address for data received by RF, RF interrupt, etc.

void set_wakeup_func(void)

Function: Wakeup setting before entering deepsleep

Parameter:

Return Value:

Note: Set output ports as internal 100K pull-down resistor, which can make sure low level can be woken up by pressing any key. Save the serial numbers of packets to the analog register 0x3b so that the serial numbers of packets will be kept after wakeup.

void package_data_init_func(void)

Function: Data initialization of RF packets

Parameter:

Return Value:

Note: Set the remote ID, rf length, etc.

unsigned char remote_key_scan_func(void)

Function: Scan key values

Parameter:

Return Value: List values of keys pressed

Note:

void main_loop(void)

Function: User main function

Parameter:

Return Value:

Note: The program loops in main_loop. When scanning key values, it sends data. The program enters suspend for 10ms after it executes a loop, which saves power consumption. The system will enter deepsleep and wait for wakeup by button press if no button is pressed within 200ms. The system will reset after wakeup.

void send_package_data_func(void)

Function: Send packets

Parameter:

Return Value:

Note: Send every packet to four frequencies to ensure the receiving frequency matches the sending frequency.