Application Note

Description of 2.4G Lighting Control Remote SDK Functions

AN-19121002-E1

Version 1.0.0

2019-12-10

Key Words:

2.4G Lighting Control Remote, SDK Function

Brief:

This document provides the description for 2.4G lighting control remote SDK functions.

TELINK SEMICONDUCTOR

Published by

Telink Semiconductor

Bldg 3, 1500 Zuchongzhi Rd,
Zhangjiang Hi-Tech Park, Shanghai, China

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.

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 (2019-12-10)

This is the initial release.

Contents

1. Overview

After a remote is powered on, the system will initialize RF, IO ports and data packets. Every remote has a 4- byte PID, which is stored in 0x3fe0 of OTP. Every time the remote send a command packet, the serial number of the packet will increase by 1.

After initialization, the system will enter normal state. The system will scan if there are keys pressed, if yes, the system will check if it is power on. If it is power on, the system will keep the group ID for later chroma or luminance adjustment. Every time the system sends a packet, the system will enter suspend for 10ms and the timer will wake it up. If there is no key pressed within 150ms, the system will enter low power mode and wait for wakeup by pressing a key.

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 control_key;
        unsigned char rf_seq_no;
        unsigned short button_keep_counter;
        unsigned short control_key_value[3];
        unsigned char reserved;
}rf_packet_led_remote_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.

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

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

```
typedef enum{
        LED_NONE_CMD=0, // Release button, send null value
        LED_ON_CMD, // Key on command
        LED_OFF_CMD, // Key off command
        LED_LUMINANCE_INC_CMD, // Luminance increase command
        LED_LUMINANCE_DEC_CMD, // Luminance decrease command
        LED_CHROME_INC_CMD, // Chroma increase command
        LED_CHROME_DEC_CMD, // Chroma decrease command
        LED_SET_CHRO_LUMI_CMD, // Set chroma/luminance command
        LED_NIGHT_CMD, // Nightlight command
        LED_LAST_CMD,
}Led_control_cmd;
```

button_keep_counter, reserved: Reserved

control_key_value: When the value of key_control is *LED_SET_CHRO_LUMI_CMD*, control_key_value[0] indicates the luminance, and control_key_value[1] indicates the chroma.

3. Definition 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 set_wakeup_func(void)

Function: Wakeup setting before entering deepsleep

Parameter:

Return Value:

Note: Set wakeup. Before IC enters low power mode, set the level of IO ports, set all output ports to low level so that wakeup pins can detect the low level when any key is pressed. Save the group ID of the remote and serial number of the packet to analog registers 0x3a and 0x3b. The two registers will reset until next power on. Previous data saved by other resets such as low power wakeup reset and watchdog reset will be kept.

void package_data_init_func(void)

Function: Initialization of RF packets

Parameter:

Return Value:

Note: Initialization of RF packets. Initialize RF packet length, read remote PIDs and VIDs, read group IDs and serial numbers of packets.

unsigned char remote_key_scan_func(void)

Function: Scan key values

Parameter:

Return Value: List values of keys pressed

Note:

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 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.