#include "stm32f10x.h"

#include "usart.h"

#include "delay.h"

#include "../BOARD/ws2812/ws2812.h"

/\*

Define a structure

C\_RED£ºRecord the value of RED LED

C\_GREEN£ºRecord the value of GREEN LED

C\_BLUE£ºRecord the value of BLUE LED

C\_WHITE£ºRecord the value of WHITE LED

C\_RED\_FLAG£ºSolve the bug of the brightness issue while re-power on the LED

\*/

struct RGB\_COLOR

{

u8 C\_RED;

u8 C\_GREEN;

u8 C\_BLUE;

u8 C\_WHITE;

u8 C\_RED\_FLAG;

u8 C\_GREEN\_FLAG;

u8 C\_BLUE\_FLAG;

};

#define ICON\_WHITE\_ADDR 0x02//Variable memory address of WHITE LED

#define ICON\_RED\_ADDR 0x03//Variable memory address of RED LED

#define ICON\_GREEN\_ADDR 0x04//Variable memory address of GREEN LED

#define ICON\_BLUE\_ADDR 0x05//Variable memory address of BLUE LED

#define TEXT\_RED\_ADDR 0x07//Variable memory address of RED LED brightness value

#define TEXT\_GREEN\_ADDR 0x08//Variable memory address of GREEN LED brightness value

#define TEXT\_BLUE\_ADDR 0x09//Variable memory address of BLUE LED brightness value

#define TEXT\_WHITE\_ADDR 0x06//Variable memory address of WHITE LED brightness value

#define SWITCH\_ONOFF\_ADDR 0x01//Variable memory address of control the ON/OFF

#define ICON\_ON 0x01//Variable memroy address of ON icon

#define ICON\_OFF 0x00//Variable memroy address of OFF icon

u8 data\_send[8]= {0xA5, 0x5A, 0x05, 0x82, 0x00, 0x00, 0x00,0x00};//The data format sent by MCU

/\*

Function sent by UART

send\_array[]£ºDATA

num£ºDATA length

\*/

void UART1\_Send\_Array(u8 send\_array[],unsigned char num)

{

u8 i=0;

while(i<num)

{

USART\_SendData(USART1,send\_array[i]);

while( USART\_GetFlagStatus(USART1,USART\_FLAG\_TC)!= SET);

i++;

}

}

/\*

Main function

\*/

int main(void)

{

uart\_init(115200); //Initiallize the baudrate as 115200

delay\_init(); //Delay the initiallization of function

struct RGB\_COLOR USER\_RGB\_COLOR;//Initiallize the RGB brightness value structure

USER\_RGB\_COLOR.C\_BLUE=0; //Initial brightness 0

USER\_RGB\_COLOR.C\_GREEN=0;

USER\_RGB\_COLOR.C\_RED=0;

USER\_RGB\_COLOR.C\_RED\_FLAG=1;

USER\_RGB\_COLOR.C\_GREEN\_FLAG=1;

USER\_RGB\_COLOR.C\_BLUE\_FLAG=1;

u16 k,q;//Count flag

u8 BLINK\_2=0;//Blink count

u8 USER\_R=0,USER\_G=0,USER\_B=0,COLOR\_TYPE=0,COLOR\_DIR=0;//Four blinking flags and brightness value

u8 blink\_type=0;//Record the single blink type

u16 times=0; //Main function delay flag

RGB\_LED\_Init(); //Initiallize the WS2812 RGB LED strip driver

while(1) //main loop

{

if(USART\_RX\_END)//Judging that serial port reception is complete

{

switch (USART\_RX\_BUF[5]) //Judging the data received is from which button

{

case 0x33: //LED number adjusting

PIXEL\_NUM=USART\_RX\_BUF[8];

break;

case LED\_ALL\_ONOFF: //LED ON/OFF

blink\_type=0;

if(USART\_RX\_BUF[8]==0)//LED OFF

{

/\*Clear the brightness and the icon of RED LED\*/

data\_send[5]=ICON\_RED\_ADDR;

data\_send[7]=ICON\_OFF;

UART1\_Send\_Array(data\_send,8);

data\_send[5]=TEXT\_RED\_ADDR;

data\_send[7]=0x00;

UART1\_Send\_Array(data\_send,8);

/\*Clear the brightness and the icon of GREEN LED\*/

data\_send[5]=ICON\_GREEN\_ADDR;

data\_send[7]=ICON\_OFF;

UART1\_Send\_Array(data\_send,8);

data\_send[5]=TEXT\_GREEN\_ADDR;

data\_send[7]=0x00;

UART1\_Send\_Array(data\_send,8);

/\*Clear the brightness and the icon of BLUE LED\*/

data\_send[5]=ICON\_BLUE\_ADDR;

data\_send[7]=ICON\_OFF;

UART1\_Send\_Array(data\_send,8);

data\_send[5]=TEXT\_BLUE\_ADDR;

data\_send[7]=0x00;

UART1\_Send\_Array(data\_send,8);

USER\_RGB\_COLOR.C\_BLUE=0;

USER\_RGB\_COLOR.C\_GREEN=0;

USER\_RGB\_COLOR.C\_RED=0;

/\*Clear the brightness and the icon of WHITE LED\*/

data\_send[5]=ICON\_WHITE\_ADDR;

data\_send[7]=ICON\_OFF;

UART1\_Send\_Array(data\_send,8);

data\_send[5]=TEXT\_WHITE\_ADDR;

data\_send[7]=0x00;

UART1\_Send\_Array(data\_send,8);

USER\_RGB\_COLOR.C\_WHITE=0;

}

else //LED ON

{

/\*Set the initial RGB value\*/

USER\_RGB\_COLOR.C\_BLUE=0x32;

USER\_RGB\_COLOR.C\_GREEN=0x10;

USER\_RGB\_COLOR.C\_RED=0x24;

USER\_RGB\_COLOR.C\_RED\_FLAG=0;

USER\_RGB\_COLOR.C\_GREEN\_FLAG=0;

USER\_RGB\_COLOR.C\_BLUE\_FLAG=0;

/\*MCU sending data to display, show the related icon\*/

data\_send[5]=ICON\_RED\_ADDR;

data\_send[7]=ICON\_ON;

UART1\_Send\_Array(data\_send,8);

data\_send[5]=TEXT\_RED\_ADDR;

data\_send[7]=0x24;

UART1\_Send\_Array(data\_send,8);

data\_send[5]=ICON\_GREEN\_ADDR;

data\_send[7]=ICON\_ON;

UART1\_Send\_Array(data\_send,8);

data\_send[5]=TEXT\_GREEN\_ADDR;

data\_send[7]=0x10;

UART1\_Send\_Array(data\_send,8);

data\_send[5]=ICON\_BLUE\_ADDR;

data\_send[7]=ICON\_ON;

UART1\_Send\_Array(data\_send,8);

data\_send[5]=TEXT\_BLUE\_ADDR;

data\_send[7]=0x32;

UART1\_Send\_Array(data\_send,8);

}

//Set LED brightness

RGB\_LED\_Write\_24Bits(USER\_RGB\_COLOR.C\_RED, USER\_RGB\_COLOR.C\_GREEN, USER\_RGB\_COLOR.C\_BLUE);

break;

case RED\_COLOR://Set RED LED brightness

blink\_type=0;

if(USER\_RGB\_COLOR.C\_RED\_FLAG==1)

{

if(USART\_RX\_BUF[8]==0)

break;

}

//Clear the WHITE LED icon

data\_send[5]=ICON\_WHITE\_ADDR;

data\_send[7]=ICON\_OFF;

UART1\_Send\_Array(data\_send,8);

//Clear the WHITE LED brightness value

data\_send[5]=TEXT\_WHITE\_ADDR;

data\_send[7]=0x00;

UART1\_Send\_Array(data\_send,8);

USER\_RGB\_COLOR.C\_WHITE=0;

//Turn on ON/OFF icon

data\_send[5]=SWITCH\_ONOFF\_ADDR;

data\_send[7]=ICON\_ON;

UART1\_Send\_Array(data\_send,8);

//Turn on RED LED icon

data\_send[5]=ICON\_RED\_ADDR;

if(USART\_RX\_BUF[8]>0)data\_send[7]=ICON\_ON;

else data\_send[7]=ICON\_OFF;

UART1\_Send\_Array(data\_send,8);

USER\_RGB\_COLOR.C\_RED=USART\_RX\_BUF[8];

USER\_RGB\_COLOR.C\_RED\_FLAG=0;

if(USER\_RGB\_COLOR.C\_RED==0)USER\_RGB\_COLOR.C\_RED\_FLAG=1;

//If the value sent from display are all 0, turn off the ON/OFF icon

if((USER\_RGB\_COLOR.C\_RED==0x00)&&(USER\_RGB\_COLOR.C\_GREEN==0x00)&&(USER\_RGB\_COLOR.C\_BLUE==0x00)&&(USER\_RGB\_COLOR.C\_WHITE==0x00))

{

data\_send[5]=SWITCH\_ONOFF\_ADDR;

data\_send[7]=ICON\_OFF;

UART1\_Send\_Array(data\_send,8);

}

RGB\_LED\_Write\_24Bits(USER\_RGB\_COLOR.C\_RED, USER\_RGB\_COLOR.C\_GREEN, USER\_RGB\_COLOR.C\_BLUE); // Red

break;

case GREEN\_COLOR://Set GREEN LED brightness

blink\_type=0;

if(USER\_RGB\_COLOR.C\_GREEN\_FLAG==1)

{

if(USART\_RX\_BUF[8]==0)

break;

}

//Turn on the GREEN LED icon

data\_send[5]=ICON\_GREEN\_ADDR;

if(USART\_RX\_BUF[8]>0)data\_send[7]=ICON\_ON;

else data\_send[7]=ICON\_OFF;

UART1\_Send\_Array(data\_send,8);

//Turn off the WHITE LED icon

data\_send[5]=ICON\_WHITE\_ADDR;

data\_send[7]=ICON\_OFF;

UART1\_Send\_Array(data\_send,8);

data\_send[5]=TEXT\_WHITE\_ADDR;

data\_send[7]=0x00;

UART1\_Send\_Array(data\_send,8);

USER\_RGB\_COLOR.C\_WHITE=0;

//Turn on the ON icon

data\_send[5]=SWITCH\_ONOFF\_ADDR;

data\_send[7]=ICON\_ON;

UART1\_Send\_Array(data\_send,8);

USER\_RGB\_COLOR.C\_GREEN=USART\_RX\_BUF[8];

USER\_RGB\_COLOR.C\_GREEN\_FLAG=0;

if(USER\_RGB\_COLOR.C\_GREEN==0)USER\_RGB\_COLOR.C\_GREEN\_FLAG=1;

if((USER\_RGB\_COLOR.C\_RED==0x00)&&(USER\_RGB\_COLOR.C\_GREEN==0x00)&&(USER\_RGB\_COLOR.C\_BLUE==0x00)&&(USER\_RGB\_COLOR.C\_WHITE==0x00))

{

data\_send[5]=SWITCH\_ONOFF\_ADDR;

data\_send[7]=ICON\_OFF;

UART1\_Send\_Array(data\_send,8);

}

RGB\_LED\_Write\_24Bits(USER\_RGB\_COLOR.C\_RED, USER\_RGB\_COLOR.C\_GREEN, USER\_RGB\_COLOR.C\_BLUE); // Green

break;

case BLUE\_COLOR://Set the BLUE LED brightness

blink\_type=0;

if(USER\_RGB\_COLOR.C\_BLUE\_FLAG==1)

{

if(USART\_RX\_BUF[8]==0)

break;

}

data\_send[5]=ICON\_BLUE\_ADDR;

if(USART\_RX\_BUF[8]>0)data\_send[7]=ICON\_ON;

else data\_send[7]=ICON\_OFF;

UART1\_Send\_Array(data\_send,8);

data\_send[5]=ICON\_WHITE\_ADDR;

data\_send[7]=ICON\_OFF;

UART1\_Send\_Array(data\_send,8);

data\_send[5]=TEXT\_WHITE\_ADDR;

data\_send[7]=0x00;

UART1\_Send\_Array(data\_send,8);

USER\_RGB\_COLOR.C\_WHITE=0;

data\_send[5]=SWITCH\_ONOFF\_ADDR;

data\_send[7]=ICON\_ON;

UART1\_Send\_Array(data\_send,8);

USER\_RGB\_COLOR.C\_BLUE=USART\_RX\_BUF[8];

USER\_RGB\_COLOR.C\_BLUE\_FLAG=0;

if(USER\_RGB\_COLOR.C\_BLUE==0)USER\_RGB\_COLOR.C\_BLUE\_FLAG=1;

if((USER\_RGB\_COLOR.C\_RED==0x00)&&(USER\_RGB\_COLOR.C\_GREEN==0x00)&&(USER\_RGB\_COLOR.C\_BLUE==0x00)&&(USER\_RGB\_COLOR.C\_WHITE==0x00))

{

data\_send[5]=SWITCH\_ONOFF\_ADDR;

data\_send[7]=ICON\_OFF;

UART1\_Send\_Array(data\_send,8);

}

RGB\_LED\_Write\_24Bits(USER\_RGB\_COLOR.C\_RED, USER\_RGB\_COLOR.C\_GREEN, USER\_RGB\_COLOR.C\_BLUE); // Blue

break;

case WHITE\_COLOR: //Set WHITE LED brightness

blink\_type=0;

data\_send[5]=ICON\_WHITE\_ADDR;

if(USART\_RX\_BUF[8]>0)data\_send[7]=ICON\_ON;

else data\_send[7]=ICON\_OFF;

UART1\_Send\_Array(data\_send,8);

data\_send[5]=ICON\_RED\_ADDR;

data\_send[7]=ICON\_OFF;

UART1\_Send\_Array(data\_send,8);

data\_send[5]=TEXT\_RED\_ADDR;

data\_send[7]=0x00;

UART1\_Send\_Array(data\_send,8);

data\_send[5]=ICON\_GREEN\_ADDR;

data\_send[7]=ICON\_OFF;

UART1\_Send\_Array(data\_send,8);

data\_send[5]=TEXT\_GREEN\_ADDR;

data\_send[7]=0x00;

UART1\_Send\_Array(data\_send,8);

data\_send[5]=ICON\_BLUE\_ADDR;

data\_send[7]=ICON\_OFF;

UART1\_Send\_Array(data\_send,8);

data\_send[5]=TEXT\_BLUE\_ADDR;

data\_send[7]=0x00;

UART1\_Send\_Array(data\_send,8);

USER\_RGB\_COLOR.C\_BLUE=0;

USER\_RGB\_COLOR.C\_GREEN=0;

USER\_RGB\_COLOR.C\_RED=0;

USER\_RGB\_COLOR.C\_RED\_FLAG=1;

USER\_RGB\_COLOR.C\_GREEN\_FLAG=1;

USER\_RGB\_COLOR.C\_BLUE\_FLAG=1;

data\_send[5]=SWITCH\_ONOFF\_ADDR;

data\_send[7]=ICON\_ON;

UART1\_Send\_Array(data\_send,8);

USER\_RGB\_COLOR.C\_WHITE=USART\_RX\_BUF[8];

if((USER\_RGB\_COLOR.C\_RED==0x00)&&(USER\_RGB\_COLOR.C\_GREEN==0x00)&&(USER\_RGB\_COLOR.C\_BLUE==0x00)&&(USER\_RGB\_COLOR.C\_WHITE==0x00))

{

data\_send[5]=SWITCH\_ONOFF\_ADDR;

data\_send[7]=ICON\_OFF;

UART1\_Send\_Array(data\_send,8);

}

RGB\_LED\_Write\_24Bits(USER\_RGB\_COLOR.C\_WHITE, USER\_RGB\_COLOR.C\_WHITE, USER\_RGB\_COLOR.C\_WHITE);

break;

case BLINK1://Blink 1

blink\_type=1;//Recording the blink type, which will be used in the loop after

//Turn off all the color icon, only ON/OFF icon remains

data\_send[5]=ICON\_RED\_ADDR;

data\_send[7]=ICON\_OFF;

UART1\_Send\_Array(data\_send,8);

data\_send[5]=TEXT\_RED\_ADDR;

data\_send[7]=0x00;

UART1\_Send\_Array(data\_send,8);

data\_send[5]=ICON\_GREEN\_ADDR;

data\_send[7]=ICON\_OFF;

UART1\_Send\_Array(data\_send,8);

data\_send[5]=TEXT\_GREEN\_ADDR;

data\_send[7]=0x00;

UART1\_Send\_Array(data\_send,8);

data\_send[5]=ICON\_BLUE\_ADDR;

data\_send[7]=ICON\_OFF;

UART1\_Send\_Array(data\_send,8);

data\_send[5]=TEXT\_BLUE\_ADDR;

data\_send[7]=0x00;

UART1\_Send\_Array(data\_send,8);

USER\_RGB\_COLOR.C\_BLUE=0;

USER\_RGB\_COLOR.C\_GREEN=0;

USER\_RGB\_COLOR.C\_RED=0;

data\_send[5]=ICON\_WHITE\_ADDR;

data\_send[7]=ICON\_OFF;

UART1\_Send\_Array(data\_send,8);

data\_send[5]=TEXT\_WHITE\_ADDR;

data\_send[7]=0x00;

UART1\_Send\_Array(data\_send,8);

USER\_RGB\_COLOR.C\_WHITE=0;

data\_send[5]=SWITCH\_ONOFF\_ADDR;

data\_send[7]=ICON\_ON;

UART1\_Send\_Array(data\_send,8);

break;

case BLINK2://Blink 2

blink\_type=2;//Recording the blink type, which will be used in the loop after

//Turn off all the color icon, only ON/OFF icon remains

data\_send[5]=ICON\_RED\_ADDR;

data\_send[7]=ICON\_OFF;

UART1\_Send\_Array(data\_send,8);

data\_send[5]=TEXT\_RED\_ADDR;

data\_send[7]=0x00;

UART1\_Send\_Array(data\_send,8);

data\_send[5]=ICON\_GREEN\_ADDR;

data\_send[7]=ICON\_OFF;

UART1\_Send\_Array(data\_send,8);

data\_send[5]=TEXT\_GREEN\_ADDR;

data\_send[7]=0x00;

UART1\_Send\_Array(data\_send,8);

data\_send[5]=ICON\_BLUE\_ADDR;

data\_send[7]=ICON\_OFF;

UART1\_Send\_Array(data\_send,8);

data\_send[5]=TEXT\_BLUE\_ADDR;

data\_send[7]=0x00;

UART1\_Send\_Array(data\_send,8);

USER\_RGB\_COLOR.C\_BLUE=0;

USER\_RGB\_COLOR.C\_GREEN=0;

USER\_RGB\_COLOR.C\_RED=0;

data\_send[5]=ICON\_WHITE\_ADDR;

data\_send[7]=ICON\_OFF;

UART1\_Send\_Array(data\_send,8);

data\_send[5]=TEXT\_WHITE\_ADDR;

data\_send[7]=0x00;

UART1\_Send\_Array(data\_send,8);

USER\_RGB\_COLOR.C\_WHITE=0;

data\_send[5]=SWITCH\_ONOFF\_ADDR;

data\_send[7]=ICON\_ON;

UART1\_Send\_Array(data\_send,8);

break;

case BLINK3://Blink 3

blink\_type=3;//Recording the blink type, which will be used in the loop after

//Turn off all the color icon, only ON/OFF icon remains

data\_send[5]=ICON\_RED\_ADDR;

data\_send[7]=ICON\_OFF;

UART1\_Send\_Array(data\_send,8);

data\_send[5]=TEXT\_RED\_ADDR;

data\_send[7]=0x00;

UART1\_Send\_Array(data\_send,8);

data\_send[5]=ICON\_GREEN\_ADDR;

data\_send[7]=ICON\_OFF;

UART1\_Send\_Array(data\_send,8);

data\_send[5]=TEXT\_GREEN\_ADDR;

data\_send[7]=0x00;

UART1\_Send\_Array(data\_send,8);

data\_send[5]=ICON\_BLUE\_ADDR;

data\_send[7]=ICON\_OFF;

UART1\_Send\_Array(data\_send,8);

data\_send[5]=TEXT\_BLUE\_ADDR;

data\_send[7]=0x00;

UART1\_Send\_Array(data\_send,8);

USER\_RGB\_COLOR.C\_BLUE=0;

USER\_RGB\_COLOR.C\_GREEN=0;

USER\_RGB\_COLOR.C\_RED=0;

data\_send[5]=ICON\_WHITE\_ADDR;

data\_send[7]=ICON\_OFF;

UART1\_Send\_Array(data\_send,8);

data\_send[5]=TEXT\_WHITE\_ADDR;

data\_send[7]=0x00;

UART1\_Send\_Array(data\_send,8);

USER\_RGB\_COLOR.C\_WHITE=0;

data\_send[5]=SWITCH\_ONOFF\_ADDR;

data\_send[7]=ICON\_ON;

UART1\_Send\_Array(data\_send,8);

break;

case BLINK4://Blink 4

blink\_type=4;//Recording the blink type, which will be used in the loop after

//Turn off all the color icon, only ON/OFF icon remains

data\_send[5]=ICON\_RED\_ADDR;

data\_send[7]=ICON\_OFF;

UART1\_Send\_Array(data\_send,8);

data\_send[5]=TEXT\_RED\_ADDR;

data\_send[7]=0x00;

UART1\_Send\_Array(data\_send,8);

data\_send[5]=ICON\_GREEN\_ADDR;

data\_send[7]=ICON\_OFF;

UART1\_Send\_Array(data\_send,8);

data\_send[5]=TEXT\_GREEN\_ADDR;

data\_send[7]=0x00;

UART1\_Send\_Array(data\_send,8);

data\_send[5]=ICON\_BLUE\_ADDR;

data\_send[7]=ICON\_OFF;

UART1\_Send\_Array(data\_send,8);

data\_send[5]=TEXT\_BLUE\_ADDR;

data\_send[7]=0x00;

UART1\_Send\_Array(data\_send,8);

USER\_RGB\_COLOR.C\_BLUE=0;

USER\_RGB\_COLOR.C\_GREEN=0;

USER\_RGB\_COLOR.C\_RED=0;

data\_send[5]=ICON\_WHITE\_ADDR;

data\_send[7]=ICON\_OFF;

UART1\_Send\_Array(data\_send,8);

data\_send[5]=TEXT\_WHITE\_ADDR;

data\_send[7]=0x00;

UART1\_Send\_Array(data\_send,8);

USER\_RGB\_COLOR.C\_WHITE=0;

data\_send[5]=SWITCH\_ONOFF\_ADDR;

data\_send[7]=ICON\_ON;

UART1\_Send\_Array(data\_send,8);

break;

default:

USART\_RX\_END=0;//Clear UART reception completion flag

USART\_RX\_STA=0;//Clear the data received by UART

break;

}

USART\_RX\_STA=0;//Clear UART reception completion flag

USART\_RX\_END=0;//Clear the data received by UART

}

else //Enter blink mode

{

if(blink\_type==1)//Function of blink type 1

{

times++;

if(times>=14)

{

times=0;

if(COLOR\_DIR==0)

{

if(COLOR\_TYPE==0)

{

USER\_R++;

USER\_G=0;

USER\_B=0;

}

else if(COLOR\_TYPE==1)

{

USER\_R=0;

USER\_G++;

USER\_B=0;

}

else if(COLOR\_TYPE==2)

{

USER\_R=0;

USER\_G=0;

USER\_B++;

}

else if(COLOR\_TYPE==3)

{

USER\_R++;

USER\_G++;

USER\_B=0;

}

else if(COLOR\_TYPE==4)

{

USER\_R=0;

USER\_G++;

USER\_B++;

}

else if(COLOR\_TYPE==5)

{

USER\_R++;

USER\_G=0;

USER\_B++;

}

if((USER\_R>=250)||(USER\_G>=250)||(USER\_B>=250))

{

COLOR\_DIR=1;

// COLOR\_TYPE++;

// if(COLOR\_TYPE>6)

// COLOR\_TYPE=0;

}

}

else

{

if(COLOR\_TYPE==0)

{

USER\_R--;

USER\_G=0;

USER\_B=0;

}

else if(COLOR\_TYPE==1)

{

USER\_R=0;

USER\_G--;

USER\_B=0;

}

else if(COLOR\_TYPE==2)

{

USER\_R=0;

USER\_G=0;

USER\_B--;

}

else if(COLOR\_TYPE==3)

{

USER\_R--;

USER\_G--;

USER\_B=0;

}

else if(COLOR\_TYPE==4)

{

USER\_R=0;

USER\_G--;

USER\_B--;

}

else if(COLOR\_TYPE==5)

{

USER\_R--;

USER\_G=0;

USER\_B--;

}

if((USER\_R==0x02)||(USER\_G==0x02)||(USER\_B==0x02))

{

COLOR\_DIR=0;

COLOR\_TYPE++;

if(COLOR\_TYPE>5)

COLOR\_TYPE=0;

}

}

RGB\_LED\_Write\_24Bits(USER\_R,USER\_G,USER\_B);

}

delay\_ms(1);

}

else if(blink\_type==2)//Blink type 2 function

{

k++;

if(k>=150)

{

k=0;

q=200;

{

BLINK\_2++;

if(BLINK\_2>8)BLINK\_2=0;

}

if(BLINK\_2==0)

RGB\_LED\_Write\_24Bits(q,0,0);

else if(BLINK\_2==1)

RGB\_LED\_Write\_24Bits(0,q,0);

else if(BLINK\_2==2)

RGB\_LED\_Write\_24Bits(0,0,q);

else if(BLINK\_2==3)

RGB\_LED\_Write\_24Bits(q,q,0);

else if(BLINK\_2==4)

RGB\_LED\_Write\_24Bits(0,q,q);

else if(BLINK\_2==5)

RGB\_LED\_Write\_24Bits(q,0,q);

else if(BLINK\_2==6)

RGB\_LED\_Write\_24Bits(q-100,q,0);

else if(BLINK\_2==7)

RGB\_LED\_Write\_24Bits(0,q-80,q);

else if(BLINK\_2==8)

RGB\_LED\_Write\_24Bits(q,0,q-120);

else if(BLINK\_2==9)

RGB\_LED\_Write\_24Bits(40,q-100,q-70);

else if(BLINK\_2==10)

RGB\_LED\_Write\_24Bits(q,100,q-80);

}

delay\_ms(1);

}

else if(blink\_type==3)//Blink type 3 function

{

k++;

if(k>=1000)

{

k=0;

{

BLINK\_2++;

if(BLINK\_2>5)BLINK\_2=0;

}

{

if(BLINK\_2==0)

RGB\_LED\_Write\_24Bits(q,0,0);

else if(BLINK\_2==1)

RGB\_LED\_Write\_24Bits(0,q,0);

else if(BLINK\_2==2)

RGB\_LED\_Write\_24Bits(0,0,q);

else if(BLINK\_2==3)

RGB\_LED\_Write\_24Bits(q,q,0);

else if(BLINK\_2==4)

RGB\_LED\_Write\_24Bits(0,q,q);

else if(BLINK\_2==5)

RGB\_LED\_Write\_24Bits(q,0,q);

}

}

delay\_ms(1);

}

else if(blink\_type==4)//Blink type 4 function

{

k++;

if(k>=500)

{

k=0;

q=0;

BLINK\_2++;

if(BLINK\_2>5)BLINK\_2=0;

}

q++;

if(q>=250)q=0;

if(BLINK\_2==0)

RGB\_LED\_Write\_24Bits(q,0,0);

else if(BLINK\_2==1)

RGB\_LED\_Write\_24Bits(0,q,0);

else if(BLINK\_2==2)

RGB\_LED\_Write\_24Bits(0,0,q);

else if(BLINK\_2==3)

RGB\_LED\_Write\_24Bits(q,q,0);

else if(BLINK\_2==4)

RGB\_LED\_Write\_24Bits(0,q,q);

else if(BLINK\_2==5)

RGB\_LED\_Write\_24Bits(q,0,q);

delay\_ms(1);

}

else //Non blink mode, means the system is not in the blink mode, no execution

{

//You may enter the related judging or execution here for the UART receiving failure

}

}

}

}