# 12-CHANNEL LED DRIVER EVALUATION BOARD GUIDE

## DESCRIPTION

The IS31FL3242 is a 12 LED current sink LED driver programmed via 1MHz I2C compatible interface. Each LED can be dimmed individually with 12-bit PWM data and each current sink has 8-bit DC scaling (Color Calibration) data which allowing 4096 steps of linear PWM dimming and 256 steps of DC current adjustable level.

Additionally each LED open and short state can be detected, IS31FL3242 store the open or short information in Open-Short Registers. The Open-Short Registers allowing MCU to read out via I2C compatible interface. Inform MCU whether there are LEDs open or short and the locations of open or short LEDs.

## FEATURES

- Supply voltage range: 2.7V to 5.5V
- 12 current sinks
- Ultra-low operational current (700 μA Typ. 900μA Max. at $V_{CC}$=3.6V)
- Accurate color rendition
  - 8/12-bit PWM/channel
  - 8-bit correction/channel
  - 8-bit global current adjust
- SDB rising edge reset I2C module
- 60kHz PWM frequency (8-bit PWM mode)
- 1MHz I2C-compatible interface
- Individual open and short error detect function
- 180 degree phase delay operation to reduce power noise
- Spread spectrum
- QFN-20 (3mm×3mm) package

## QUICK START


*Figure 1: Photo of IS31FL3242 Evaluation Board*

## RECOMMENDED EQUIPMENT

- 5.0V, 2A power supply

## ABSOLUTE MAXIMUM RATINGS

- ≤ 5.5V power supply

***Caution: Do not exceed the conditions listed above, otherwise the board will be damaged.***

## PROCEDURE

The IS31FL3242 evaluation board is fully assembled and tested. Follow the steps listed below to verify board operation.

***Caution: Do not turn on the power supply until all connections are completed*.**

1) If using external DC power supply connect the ground terminal of the power supply to the evaluation board's GND pin and the positive terminal to the VCC pin. The evaluation board can also be powered via the Micro USB connector.
2) Short JP6 to close external control.
3) Open JP5 to pull AD pin down to GND.
4) Turn on the power supply/Plug in the Micro USB and pay attention to the supply current. If the current exceeds 1A, please check for circuit fault.
5) Enter the desired mode of display by toggling the MODE button (K1).

## EVALUATION BOARD OPERATION

The IS31FL3242 evaluation board has five display modes. Press K1 to switch configurations:

***Note: See Appendix for each mode's detail.***

1) Two groups single color LEDs chasing each other-A.
2) Two groups single color LEDs chasing each other-B.
3) Three groups single color LEDs chasing after each other.
4) Chasing cycle.
5) RGB LEDs (D19-D22) are breathing effect A-mixed color.
6) RGB LEDs (D19-D22) are breathing effect B.

***Note: IS31FL3242 solely controls the FxLED function on the evaluation board.***

## ORDERING INFORMATION

| Part No. | Temperature Range | Package |
|---|---|---|
| IS31FL3242-QFLS4-EB | -40°C to +125°C, Industrial | QFN-20, Lead-free |

*Table 1: Ordering Information*

***For pricing, delivery, and ordering information, please contacts Lumissil's analog marketing team at analog@Lumissil.com or (408) 969-6600.***

## SOFTWARE SUPPORT

JP6 (EXT CTRL) default setting is closed (jumper on). If it is open (when the EVB is powered on by 5V DC or micro-USB, no jumper JP6), the on-board MCU will configure its own I2C/SDB pins to High Impedance status so an external source can driver the I2C/SDB signals to control the IS31FL3242 LED driver, the on-board MCU will also configure the U4 to open the VLED (Single color LED+) and close the VRGB. When JP5 open, AD pin pulled down to GND by R13.

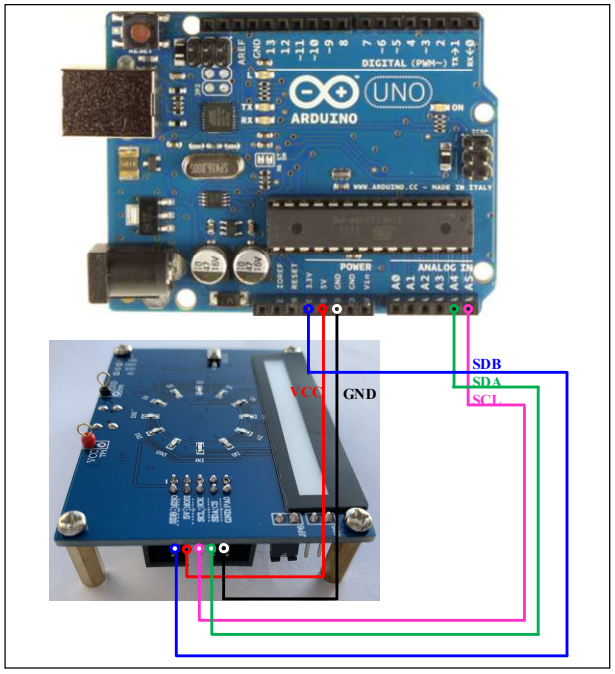


***Figure 2: Photo of Arduino UNO connected to Evaluation Board***

The steps listed below are an example using the Arduino for external control.

The Arduino hardware consists of an Atmel microcontroller with a bootloader allowing quick firmware updates. First download the latest Arduino Integrated Development Environment IDE (1.6.12 or greater) from www.arduino.cc/en/Main/Software. Also download the Wire.h library from www.arduino.cc/en/reference/wire and verify that pgmspace.h is in the directory …program Files(x86)/Arduino/hardware/tools/avr/avr/include/avr/. Then download the latest IS31FL3242 test firmware (sketch) from the Lumissil website http://www.lumissil.com/products/led-driver/fxled.

1) Open JP5 and JP6.
2) Connect the 5 pins from Arduino board to IS31FL3242 EVB:
   a) Arduino 5V pin to IS31FL3242 EVB VCC.
   b) Arduino GND to IS31FL3242 EVB GND.
   c) Arduino SDA (A4) to IS31FL3242 EVB SDA.
   d) Arduino SCL (A5) to IS31FL3242 EVB SCL.
   e) If Arduino use 3.3V MCU VCC, connect 3.3V to IS31FL3242 EVB SDB, if Arduino use 5.0V MCU VCC, connect 5.0V to EVB SDB.
   (Arduino UNO is 5.0V, so SDB=5.0V)
3) Use the test code in appendix I or download the test firmware (sketch) from the Lumissil website, a .txt file and copy the code to Arduino IDE, compile and upload to Arduino.
4) Run the Arduino code and the initial mode will change the Blue LED brightness every second. Note: the white color LEDs cannot be controlled when the onboard STM32F103C8T6 is disabled.
   (Some early board we provided is still controlling the single color LED, if want to switch between single color LED and RGB, remove the U4 and connect the VCC to the LED+)

***Please refer to the datasheet to get more information about IS31FL3242.***

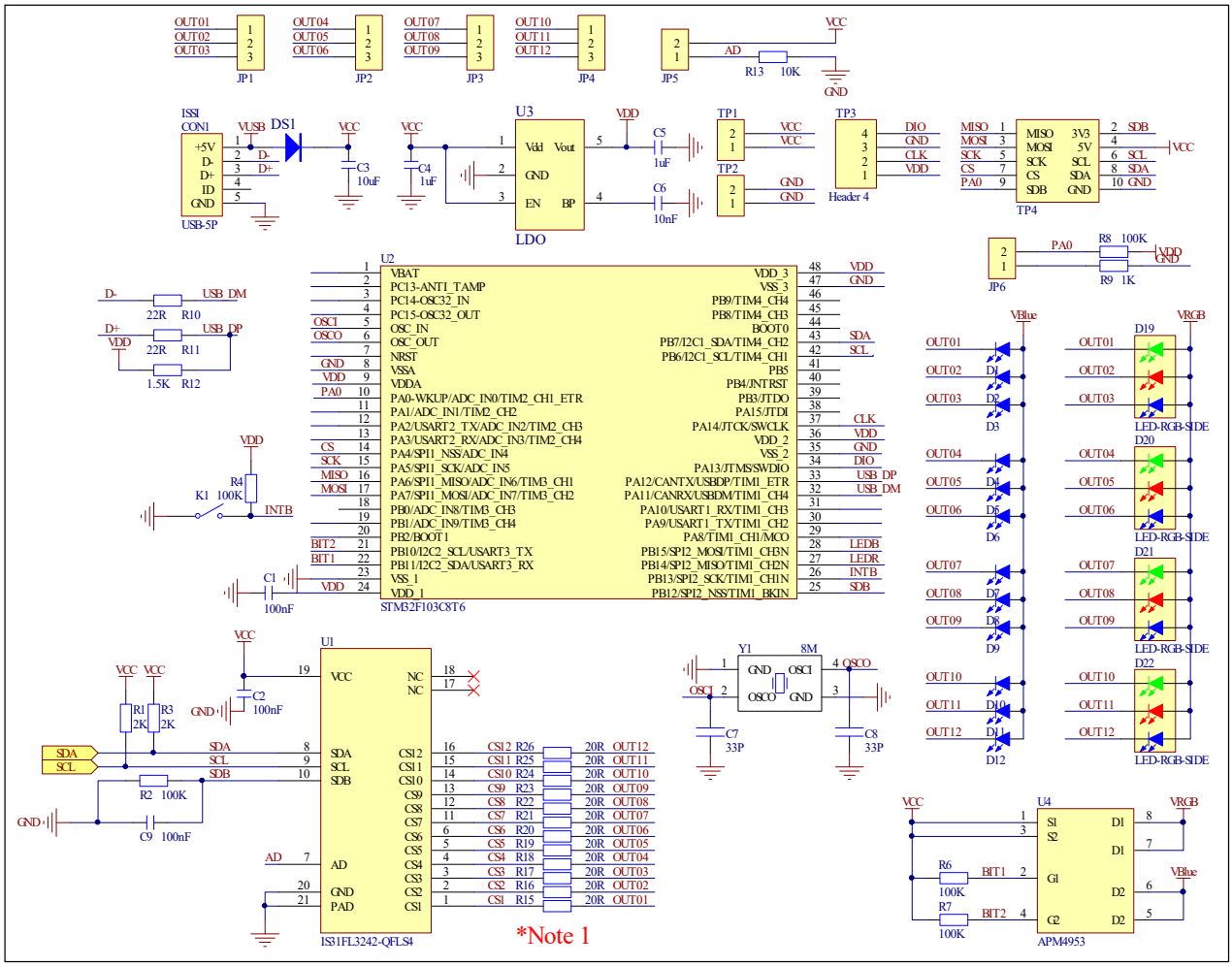


*Figure 3: IS31FL3242 Application Schematic*

## BILL OF MATERIALS

| Name | Symbol | Description | Qty | Supplier | Part No. |
|---|---|---|---|---|---|
| LED Driver | U1 | Matrix LED Driver | 1 | Lumissil | IS31FL3242 |
| MCU | U2 | Microcontroller | 1 | STM | STM32F103C8T6 |
| LDO | U3 | 3.0V LDO | 1 | SGMICRO | SGM2019-3.0YN5G |
| PMOS | U4 | PMOS | 1 | ANPEC | APM4953 |
| LED | D1~D12 | LED, SMD Blue | 12 | EVERLIGHT | 19-217/BHC-AN1P2/3T |
| RGB LED | D19~D22 | RGB LED, SMD | 4 | EVERLIGHT | 99-235/RSGBB7C-A22/2D or 99-235/RGBC/TR8 |
| Diode | DS1 | Diode, SMD | 1 | DIODES | DFLS240 |
| Crystal | Y1 | Crystal, 8MHz | 1 | HLX | HC-49S |
| Resistor | R1,R3 | RES,2k,1/10W,±5%,SMD | 2 | Yageo | RC0603JR-072KL |
| Resistor | R2,R4, R6,R7,R8 | RES,100k,1/10W,±5%,SMD | 5 | Yageo | RC0603JR-07100KL |
| Resistor | R9 | RES,1k,1/10W,±5%,SMD | 1 | Yageo | RC0603JR-071KL |
| Resistor | R10,R11 | RES,22R,1/10W,±5%,SMD | 2 | Yageo | RC0603JR-0722RL |
| Resistor | R12 | RES,1.5k,1/10W,±5%,SMD | 1 | Yageo | RC0603JR-071K5L |
| Resistor | R13 | RES,10k,1/10W,±5%,SMD | 1 | Yageo | RC0603JR-07100KL |
| Resistor | R16,R19,R22, R25 | RES,20R,1/10W,±5%,SMD (Note 1) | 4 | Yageo | RC0603JR-0720RL |
| Resistor | R15,R17,R18, R20,R21,R23, R24,R26 | RES,20R,1/10W,±5%,SMD | 8 | Yageo | RC0603JR-0720RL |
| Capacitor | C1,C2,C9 | CAP,100nF,16V,±20%,SMD | 3 | Yageo | CC0603MRX7R7BB104 |
| Capacitor | C3 | CAP,10μF,16V,±20%,SMD | 1 | Yageo | CC0603MRX5R7BB106 |
| Capacitor | C4,C5 | CAP, 1μF,16V,±10%,SMD | 2 | Yageo | CC0603KRX7R7BB105 |
| Capacitor | C6 | CAP,10nF,16V,±10%,SMD | 1 | Yageo | CC0603KPX7R7BB103 |
| Capacitor | C7,C8 | CAP,33pF,50V,±5%,SMD | 2 | Yageo | CQ0603JRNPO9BN360 |
| Button | K1(Bottom) | Button | 1 | | |
| Micro USB | CON1 | Micro USB | 1 | | |

***Bill of Materials, refer to Figure 3 above.***

***Note 1: The value of these resistors on the evaluation board is 20Ω. For $PV_{CC}$=5V and red LED application, prefer 51Ω for these resistors as shown in datasheet Figure 1.***

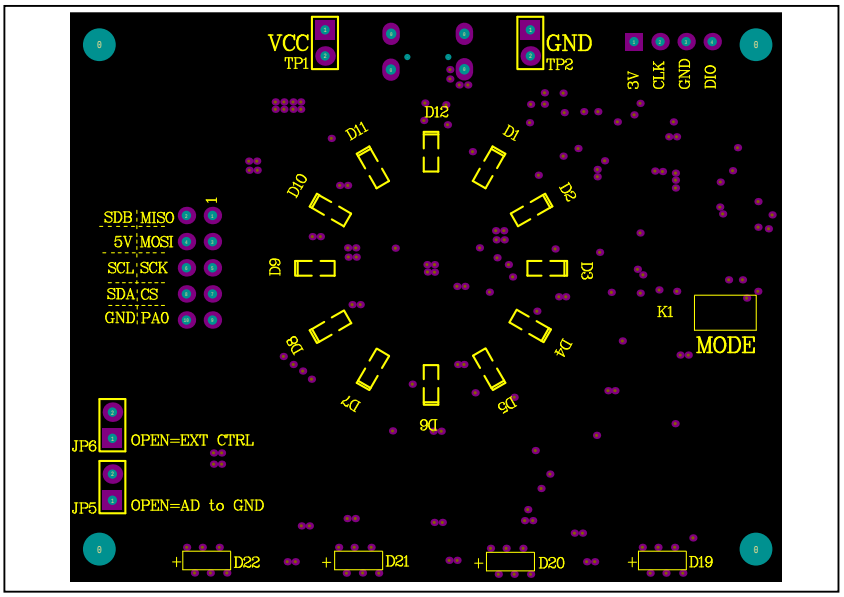


*Figure 4: Board Component Placement Guide - Top Layer*

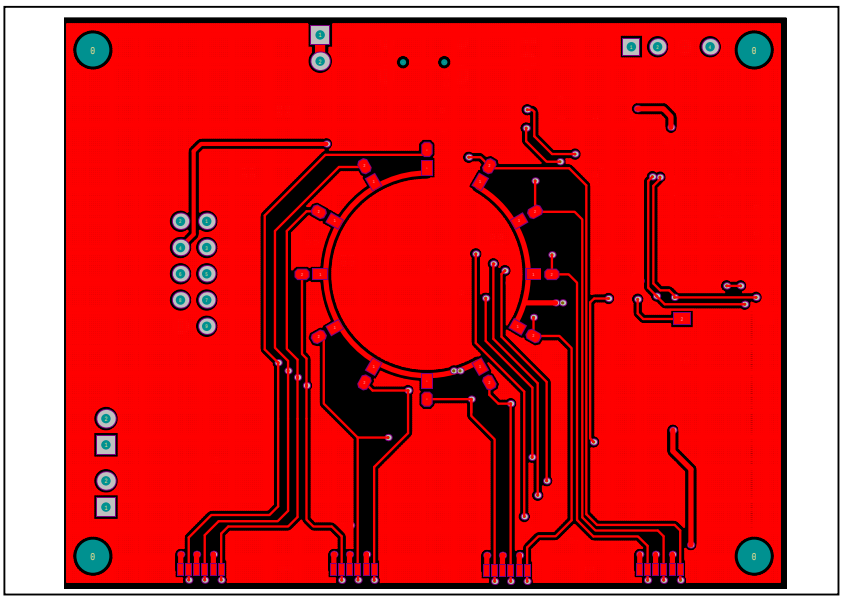

*Figure 5: Board PCB Layout - Top Layer*

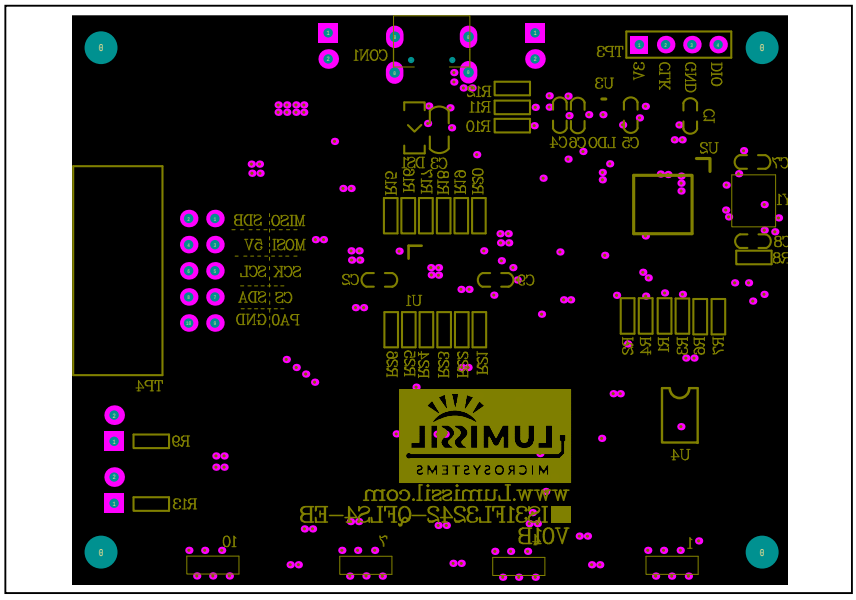

*Figure 6: Board Component Placement Guide - Bottom Layer*

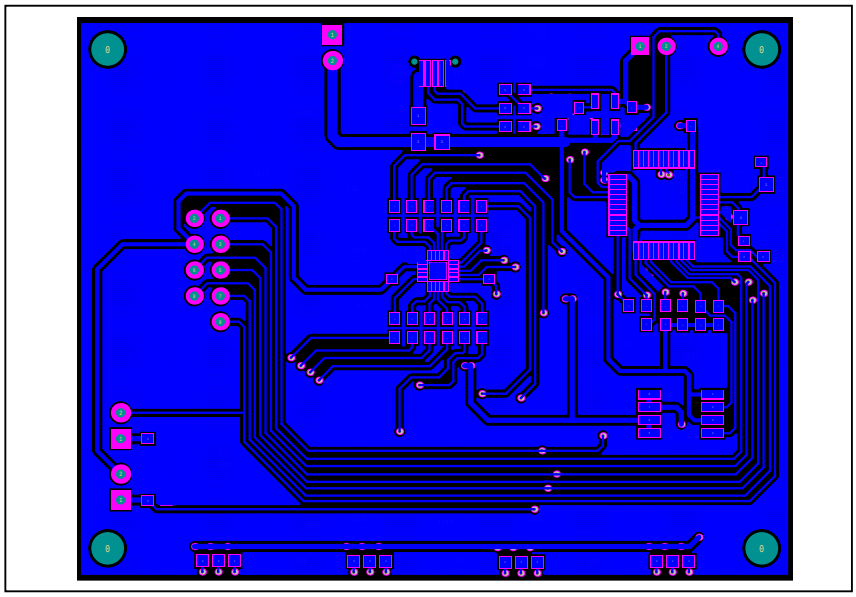

*Figure 7: Board PCB Layout - Bottom Layer*

## REVISION HISTORY

| Revision | Detail Information | Date |
|---|---|---|
| A | Initial release | 2020.11.30 |
| B | Update the BOM | 2021.06.04 |

## APPENDIX Ⅰ: IS31FL3242 Arduino Test Code V01A

```
#include<Wire.h>
#include<avr/pgmspace.h>
#define Addr_GND 0x88


int PWM_Gama64[128]=
{
  0x00,0x01,0x02,0x03,0x04,0x05,0x06,0x07,
  0x08,0x09,0x0b,0x0d,0x0f,0x11,0x13,0x16,
  0x1a,0x1c,0x1d,0x1f,0x22,0x25,0x28,0x2e,
  0x34,0x38,0x3c,0x40,0x44,0x48,0x4b,0x4f,
  0x55,0x5a,0x5f,0x64,0x69,0x6d,0x72,0x77,
  0x7d,0x80,0x88,0x8d,0x94,0x9a,0xa0,0xa7,
  0xac,0xb0,0xb9,0xbf,0xc6,0xcb,0xcf,0xd6,
  0xe1,0xe9,0xed,0xf1,0xf6,0xfa,0xfe,0xff,

  0xff,0xfe,0xfa,0xf6,0xf1,0xed,0xe9,0xe1,
  0xd6,0xcf,0xcb,0xc6,0xbf,0xb9,0xb0,0xac,
  0xa7,0xa0,0x9a,0x94,0x8d,0x88,0x80,0x7d,
  0x77,0x72,0x6d,0x69,0x64,0x5f,0x5a,0x55,
  0x4f,0x4b,0x48,0x44,0x40,0x3c,0x38,0x34,
  0x2e,0x28,0x25,0x22,0x1f,0x1d,0x1c,0x1a,
  0x16,0x13,0x11,0x0f,0x0d,0x0b,0x09,0x08,
  0x07,0x06,0x05,0x04,0x03,0x02,0x01,0x00
};

void setup() {
  // put your setup code here, to run once:
  Wire.begin();
  Wire.setClock(400000);//I2C 400kHz
//  pinMode(4,OUTPUT);//SDB
//  digitalWrite(4,HIGH);//SDB_HIGH
    Init_FL3242();
}

void loop() {
  // put your main code here, to run repeatedly:
   //IS31FL3242_mode1();//8-bit mode
   IS31FL3242_mode2();//12-bit mode
}

void IS_IIC_WriteByte(uint8_t Dev_Add,uint8_t Reg_Add,uint8_t Reg_Dat)
{
  Wire.beginTransmission(Dev_Add/2);
```

```
  Wire.write(Reg_Add); // sends regaddress
  Wire.write(Reg_Dat); // sends regaddress
  Wire.endTransmission(); // stop transmitting
}

void Init_FL3242(void)
{
  int i = 0;
  IS_IIC_WriteByte(Addr_GND,0x00,0x33);//Enable SSD 8-bit mode: 0x73 12-bit mode: 0x33
  IS_IIC_WriteByte(Addr_GND,0x01,0x80);//GCC
  IS_IIC_WriteByte(Addr_GND,0x61,0x00);//frequency select
  for(i=0x0E;i<=0x25;i++)
  {
      IS_IIC_WriteByte(Addr_GND,i,0x00);//PWM
  }
  for(i=0x02;i<=0x0D;i++)
  {
      IS_IIC_WriteByte(Addr_GND,i,0xFF);//SL
  }
}

void IS31FL3242_mode1(void)// 8-bit mode
{
  int   i = 0;
  int   j = 0;
   for (j=1;j<=127;j++)//all LED breath falling
  {
     for(i=0x0E;i<=0x24;i=i+2)
    {
      IS_IIC_WriteByte(Addr_GND,i,PWM_Gama64[j]);//set all PWM
    }
    delay(20);//20ms
  }
}

void IS31FL3242_mode2(void)// 12-bit mode
{
  int   i, j,k,l;

  for(k=0;k<=0x0f;k++)//all LED breath rising
  {
    for(l=0x0F;l<=0x25;l=l+2)
    {
      IS_IIC_WriteByte(Addr_GND,l,k);//set all PWM
    }
    for (j=0;j<=255;j++)//all LED breath falling
```

```
    {
      for(i=0x0E;i<=0x24;i=i+2)
      {
        IS_IIC_WriteByte(Addr_GND,i,j);//set all PWM
      }
      delay(2);//2ms
    }
  }

    for(k=0x0f;k>=0;k--)//all LED breath falling
  {
    for(l=0x0F;l<=0x25;l=l+2)
    {
      IS_IIC_WriteByte(Addr_GND,l,k);//set all PWM
    }
    for (j=255;j>=0;j--)//all LED breath falling
    {
      for(i=0x0E;i<=0x24;i=i+2)
      {
        IS_IIC_WriteByte(Addr_GND,i,j);//set all PWM
      }
      delay(2);//2ms
    }
  }
}
```