## DESCRIPTION

The IS31FL3746B is a general purpose 18×n (n=1~4) LED Matrix programmed via 12MHz SPI interface. Each LED can be dimmed individually with 8-bit PWM data and 8-bit DC scaling (Color Calibration) data which allowing 256 steps of linear PWM dimming and 256 steps of DC current adjustable level.

Additionally each LED open and short state can be detected, IS31FL3746B store the open or short information in Open-Short Registers. The Open-Short Registers allowing MCU to read out via SPI 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
- 18 current sinks
- Support 18×n (n=1~4) LED matrix configurations
- Individual 256 PWM control steps
- Individual 256 DC current steps
- Global 256 current steps
- SDB rising edge reset SPI module
- 29kHz PWM frequency
- 12MHz SPI interface
- State lookup registers
- Individual open and short error detect function
- De-Ghost
- QFN-32 (4mm×4mm) package

## QUICK START

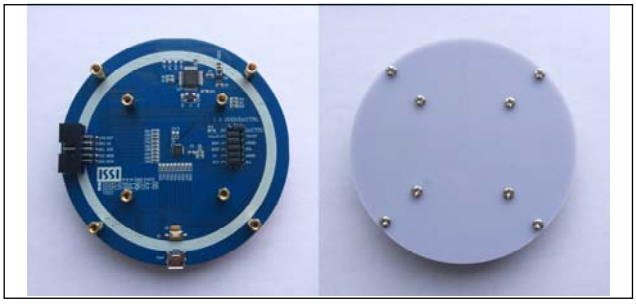


*Figure 1: Photo of IS31FL3746B Evaluation Board*

## 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 IS31FL3746B 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) Connect Pin1 and Pin2 in JP6 to enable the control of board MCU (default status).
2) Connect Pin2 and Pin3 in JP5 to connect the VIO to 3V.
3) Connect Pin1 and Pin2 in JP1 (MISO) / JP2 (MOSI) / JP3 (CS) / JP4 (SCK).
4) Connect the 5VDC power to VCC/GND of TP4, or plug in the USB power input to micro-USB.
5) Turn on the power supply, pay attention to the supply current. If the current exceeds 1A, please check for circuit fault.

## EVALUATION BOARD OPERATION

The IS31FL3746B evaluation board has three display modes. Press K1 to switch configurations:

1) (Default mode) Colors breathe change.
2) Rainbow.
3) Half cycle colors change.

***Note: IS31FL3746B solely controls the FxLED function on the evaluation board.***

## ORDERING INFORMATION

| Part No. | Temperature Range | Package |
|---|---|---|
| IS31FL3746B-QFLS4-EB | -40°C to +125°C, Industrial | QFN-32, 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

Pin1 and Pin2 of JP6 is default shorted. If the jumper is opened or move to Pin 2 and Pin 3, the on-board MCU will configure SPI pins and SDB pin to high impedance. External SPI and SDB signals can be connected to TP4 (Above Lumissil Logo) to control the IS31FL3746B LED driver.

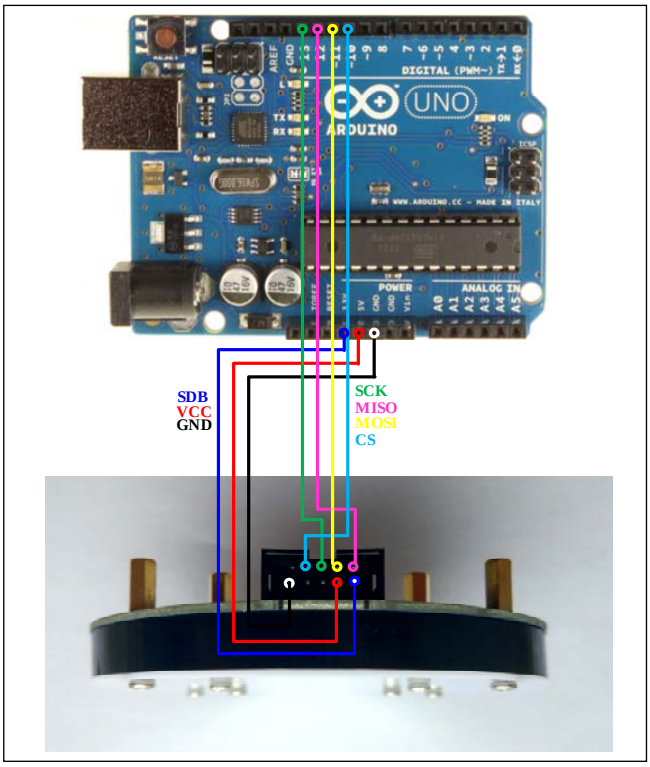



***Figure 2: Photo of Arduino UNO connected to Evaluation Board***

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 IS31FL3746B test firmware (sketch) from the Lumissil website http://www.lumissil.com/products/led-driver/fxled.

1) Open JP6's jumper or move to Pin 2 and Pin 3(Pin 3 is floated).
2) Move JP5's jumper to Pin 1 and Pin 2(VIO connect to 5V).
3) Connect the 7 pins from Arduino board to TP4(Above Lumissil Logo):
   a) Arduino 5V to IS31FL3746B EVB GND.
   b) Arduino GND to IS31FL3746B EVB GND.
   c) Arduino 3.3V pin to IS31FL3746B EVB SDB.
   d) Arduino SCK (13) to IS31FL3746B EVB SCK.
   e) Arduino MISO (12) to IS31FL3746B EVB MISO.
   f) Arduino MOSI (11) to IS31FL3746B EVB MOSI.
   g) Arduino SS (10) to IS31FL3746B EVB CS.
   h) If Arduino use 3.3V MCU VCC, connect 3.3V to IS31FL3746B EVB SDB, if Arduino use 5.0V MCU VCC, connect 5.0V to EVB SDB.
   (Arduino UNO MCU VCC is 5V, so SDB can be 5V or 3.3V)
4) 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.
5) Run the Arduino code and the initial mode will change the RGB LED colors.

***Please refer to the datasheet to get more information about IS31FL3746B.***

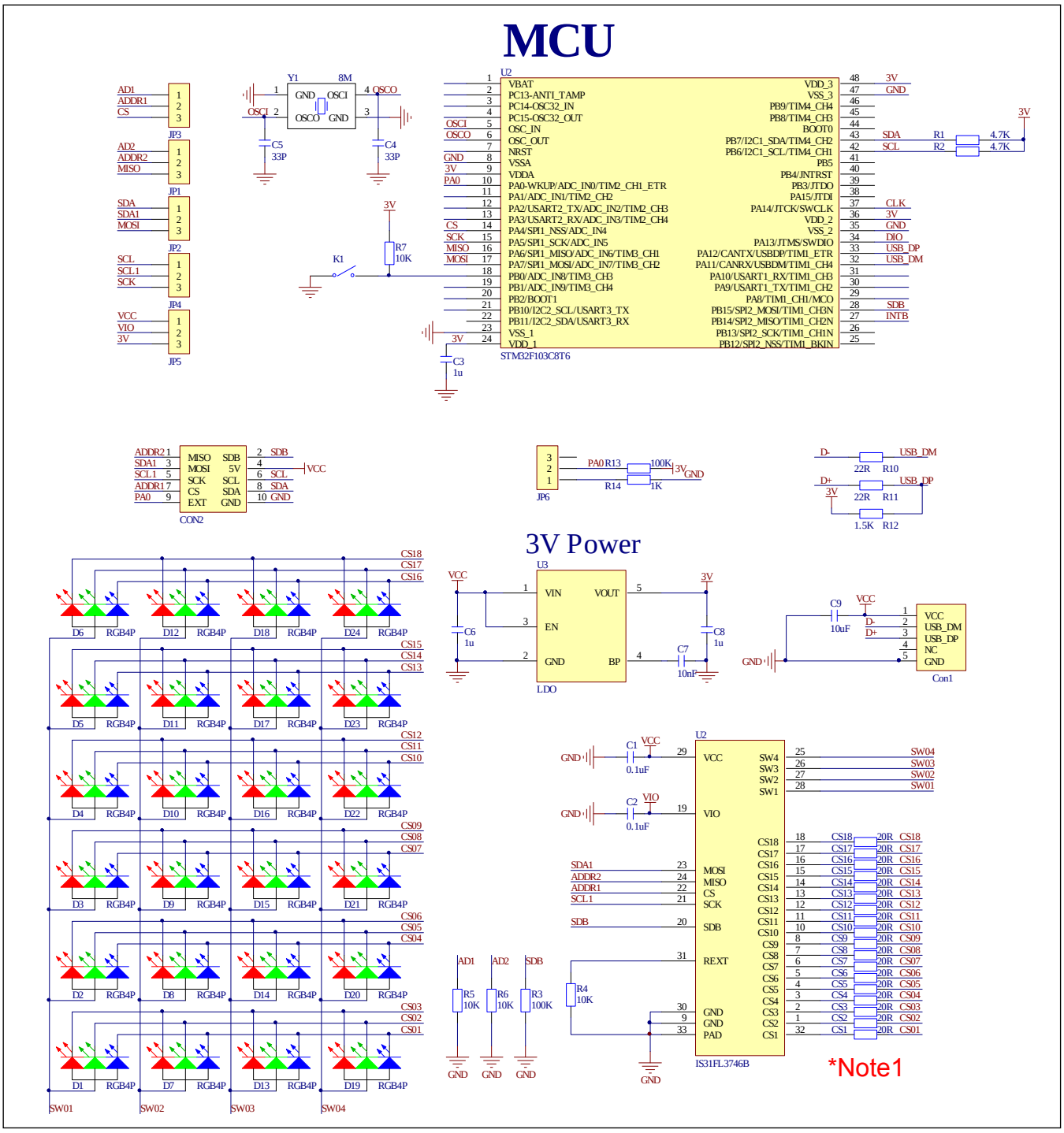



*Figure 3: IS31FL3746B Application Schematic*

## BILL OF MATERIALS

| Name | Symbol | Description | Qty | Supplier | Part No. |
|---|---|---|---|---|---|
| LED Driver | U1 | Matrix LED Driver | 1 | Lumissil | IS31FL3746B |
| MCU | U2 | Microcontroller | 1 | STM | STM32F103C8T6 |
| LDO | U3 | Reduced voltage | 1 | SGMICRO | SGM2019-3.3V |
| Diode | D1~D24 | RGB LED, SMD | 24 | Everlight | 9-237/R6GHBHC-A01/2T |
| Crystal | Y1 | Crystal, 8MHz | 1 | JB | HC-49S |
| Resistor | R1,R2 | RES,4.7k,1/10W,±5%,SMD | 2 | Yageo | RC0603JR-074K7L |
| Resistor | R3,R13 | RES,100k,1/10W,±5%,SMD | 2 | Yageo | RC0603JR-07100KL |
| Resistor | R4,R5,R6,R7 | RES,10k,1/10W,±5%,SMD | 4 | Yageo | RC0603JR-0710KL |
| 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 | R14 | RES,1k,1/10W,±5%,SMD | 1 | Yageo | RC0603JR-071KL |
| Resistor | CS1,CS2,CS4, CS5,CS7,CS8, CS10,CS11,CS13, CS14,CS16,CS17 | RES,20R,1/10W,±5%,SMD | 12 | Yageo | RC0603JR-0720RL |
| Resistor | CS3,CS6,CS9, CS12,CS15,CS18 | RES,20R,1/10W,±5%,SMD (Note 1) | 6 | Yageo | RC0603JR-0720RL |
| Capacitor | C1,C2 | CAP,100nF,16V,±20%,SMD | 2 | Yageo | CC0603MRX7R7BB104 |
| Capacitor | C3,C6,C8 | CAP, 1μF,16V,±10%,SMD | 3 | Yageo | CC0603KRX7R7BB105 |
| Capacitor | C4,C5 | CAP,33pF,50V,±5%,SMD | 2 | Yageo | CQ0603JRNPO9BN330 |
| Capacitor | C7 | CAP,10nF,16V,±10%,SMD | 1 | Yageo | CC0603KRX7R7BB103 |
| Capacitor | C9 | CAP,10μF,16V,±20%,SMD | 1 | Yageo | CC0603MRX5R7BB106 |
| Button | K1 | Button SMD | 1 | | |

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

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

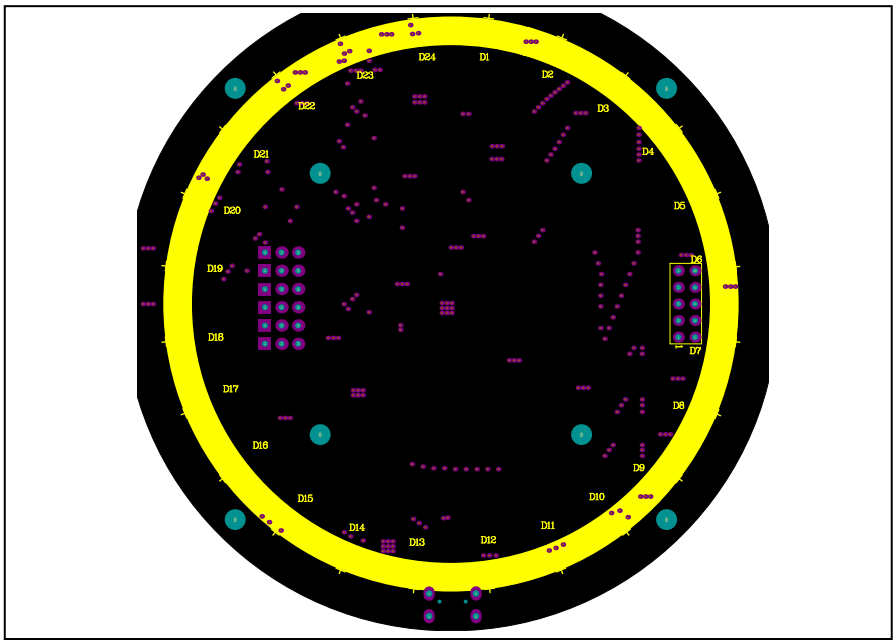



*Figure 5: Board Component Placement Guide - Top Layer*

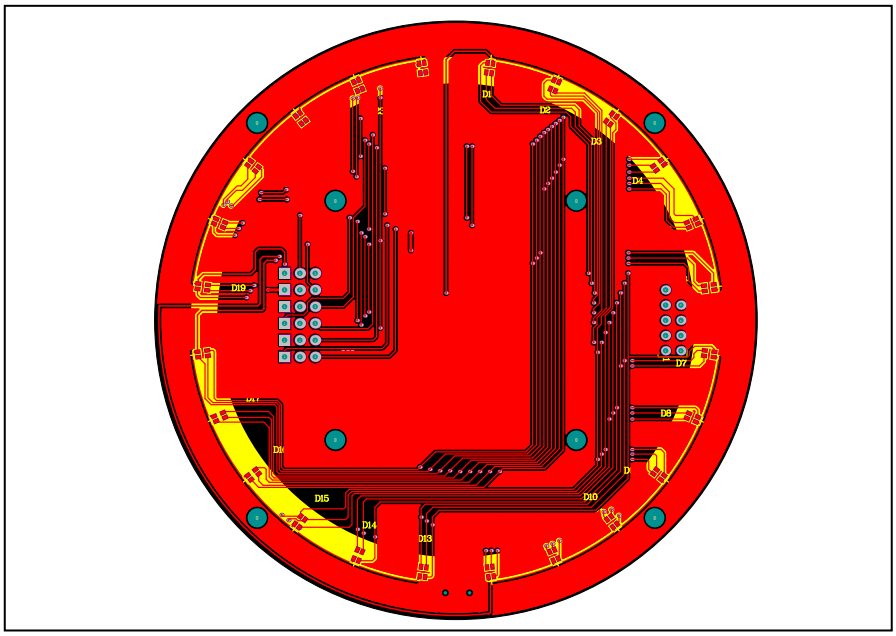


*Figure 6: Board PCB Layout - Top Layer*

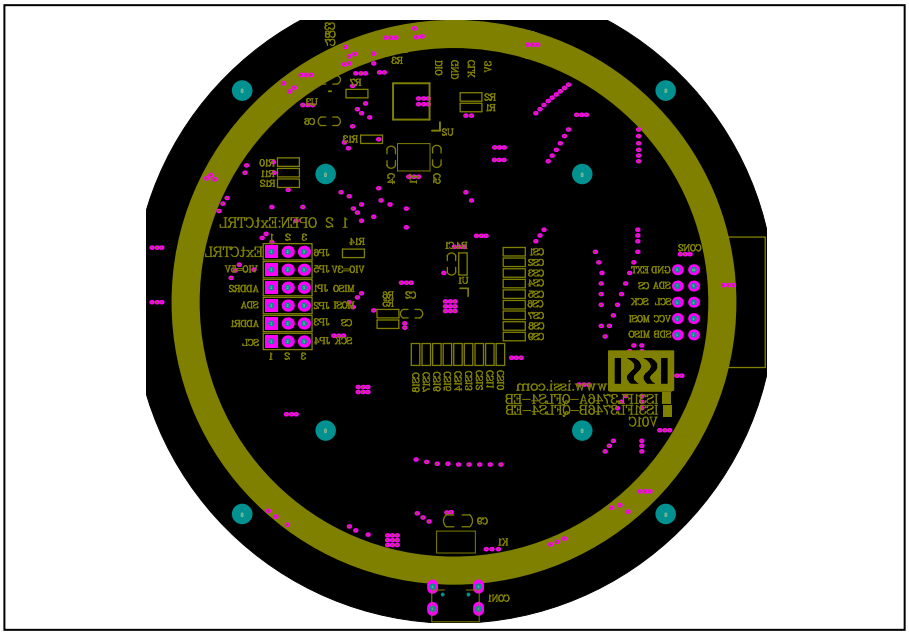


*Figure 7: Board Component Placement Guide - Bottom Layer*

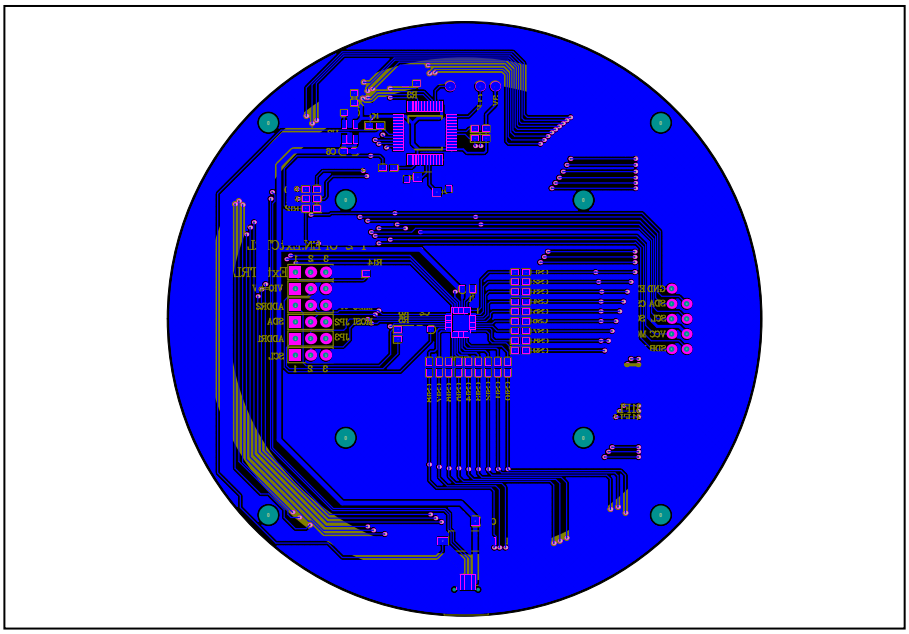


*Figure 8: Board PCB Layout - Bottom Layer*

Copyright © 2018 Lumissil Microsystems. All rights reserved. Lumissil Microsystems reserves the right to make changes to this specification and its products at any time without notice. Lumissil Microsystems assumes no liability arising out of the application or use of any information, products or services described herein. Customers are advised to obtain the latest version of this device specification before relying on any published information and before placing orders for products.
Lumissil Microsystems does not recommend the use of any of its products in life support applications where the failure or malfunction of the product can reasonably be expected to cause failure of the life support system or to significantly affect its safety or effectiveness. Products are not authorized for use in such applications unless Lumissil Microsystems receives written assurance to its satisfaction, that:
a.) the risk of injury or damage has been minimized;
b.) the user assume all such risks; and
c.) potential liability of Lumissil Microsystems is adequately protected under the circumstances

## REVISION HISTORY

| Revision | Detail Information | Date |
|---|---|---|
| A | Initial release | 2018.12.19 |

## APPENDIX Ⅰ: IS31FL3746B Arduino Test Code V01A

```
#include<SPI.h>
#include<avr/pgmspace.h>

#define Addr_Write_Page0 0x40
#define Addr_Write_Page1 0x41
const int slaveSelectPin = 10;

byte PWM_Gamma64[64]=
{
  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
};


void setup()
{
  // put your setup code here, to run once:
  // set the slaveSelectPin as an output:
  pinMode (slaveSelectPin, OUTPUT);
  // initialize SPI:
  SPI.begin();
  SPI.beginTransaction(SPISettings(20000000, MSBFIRST, SPI_MODE0));
  //SPI.setClockDivider(SPI_CLOCK_DIV4);
  SPI.setDataMode(3);
}

void loop() {
  // put your main code here, to run repeatedly:
  mainloop();
}

void SPI_WriteByte(uint8_t Dev_Add,uint8_t Reg_Add,uint8_t Reg_Dat)   //writing an LED register
{
  digitalWrite(slaveSelectPin, LOW);     // take the SS pin low to select the chip:
  SPI.transfer(Dev_Add);     //    send in the address and value via SPI:
  SPI.transfer(Reg_Add);
  SPI.transfer(Reg_Dat);
  digitalWrite(slaveSelectPin, HIGH);     // take the SS pin high to de-select the chip:
}
uint8_t SPI_ReadByte(uint8_t address)
{
  digitalWrite(slaveSelectPin,LOW);
  SPI.transfer(0x05);
  SPI.transfer(address);
  uint8_t result = SPI.transfer(0x00);
  digitalWrite(slaveSelectPin, HIGH);
  return result;
 }
void Init3746B(void)
{
  int i,j;
  for(i=0;i<0x48;i++)
  {
    SPI_WriteByte(Addr_Write_Page0,i,0);//PWM
  }

  for(i=1;i<0x49;i++)
  {
    SPI_WriteByte(Addr_Write_Page1,i,0xff);//scaling
  }

  SPI_WriteByte(Addr_Write_Page1,0x52,0x70);
  SPI_WriteByte(Addr_Write_Page1,0x51,0xff);//GCC
  SPI_WriteByte(Addr_Write_Page1,0x50,0x09);//
}
```

```
void mainloop(void)//
{
   int i,j;
   Init3746B();
   //   digitalWrite(0, LOW);      // turn the ARDUINO BOARD LED on (HIGH is the voltage level)
   while(1)
   {
      for(j=0;j<64;j++)//BLUE
      {
         for(i=1;i<0x48;i=i+3)
         {
            SPI_WriteByte(Addr_Write_Page0,i,PWM_Gamma64[j]);//PWM}
         }
       }
       delay(500);

      for(j=63;j>=0;j--)
      {
         for(i=1;i<0x48;i=i+3)
         {
            SPI_WriteByte(Addr_Write_Page0,i,PWM_Gamma64[j]);//PWM}
         }
       }
       delay(500);
      for(j=0;j<64;j++)//GREEN
      {
         for(i=2;i<0x48;i=i+3)
         {
            SPI_WriteByte(Addr_Write_Page0,i,PWM_Gamma64[j]);//PWM}
         }
       }
       delay(500);

      for(j=63;j>=0;j--)
      {
         for(i=2;i<0x48;i=i+3)
         {
            SPI_WriteByte(Addr_Write_Page0,i,PWM_Gamma64[j]);//PWM}
         }
       }
       delay(500);

      for(j=0;j<64;j++)//RED
      {
         for(i=3;i<0x49;i=i+3)
         {
            SPI_WriteByte(Addr_Write_Page0,i,PWM_Gamma64[j]);//PWM}
         }
       }
       delay(500);

      for(j=63;j>=0;j--)
      {
         for(i=3;i<0x49;i=i+3)
         {
            SPI_WriteByte(Addr_Write_Page0,i,PWM_Gamma64[j]);//PWM}
         }
       }
       delay(500);


      for(j=0;j<64;j++)//WHITE
      {
         for(i=1;i<0x49;i++)
         {
            SPI_WriteByte(Addr_Write_Page0,i,PWM_Gamma64[j]);//PWM}
         }
       }
       delay(500);

      for(j=63;j>=0;j--)
      {
         for(i=1;i<0x49;i++)
         {
            SPI_WriteByte(Addr_Write_Page0,i,PWM_Gamma64[j]);//PWM}
```

```
        }
      }
      delay(500);
  }
}
```